

Treball Final de Grau

***Disseny d'una estació
automatitzada de control
de nivell, pressió i cabal***

Grau en Enginyeria Electrónica Industrial i Automàtica

Curs 17/18

Autor: . DANIEL ROMERO RODRIGUEZ

Director: Teresa Escobet Canal i David Soler Jimenez

Data: 06-07-2018

Localitat: Manresa

Aquest treball s'ha pogut realitzar gràcies al suport dels nostres tutors Teresa Escobet i David Soler que han estat molt atents durant el transcurs del present treball.

RESUM DEL PROJECTE (en català o castellà)

El projecte neix de la necessitat de conèixer més a fons el món de l'automatització i el control d'un sistema automatitzat fent servir elements d'una estació antiga que incorpora transductors analògics i digitals .

Durant el transcurs del treball, es desmunta una antiga maqueta definida per cinc estacions i es reconstrueix una nova maqueta que constà d'una sola estació automatitzada en la qual es pot exercir un control de nivell, pressió i/o cabal.

Pel control de la nova estació s'utilitza un PLC modular de l'empresa *B&R Industrial Automation GmbH* de la sèrie X20, concretament el X20CP1381, utilitzant el software *Automation Studio*.

En el projecte es plantegen diferents programes de control de pressió de l'estació utilitzant el llenguatge de programació Text Estructurat . En els programes s'utilitza la guia GEMMA i s'implementa un controlador PID en el programa de regulació de pressió.

RESUM DEL PROJECTE (en anglès)

The project grows from the need to learn more about the world of automation and the control of an automated system using elements from an old station that incorporates analog and digital transducers.

During the course of the work, an old mock-up defined by five stations is disassembled and a new mock-up is reconstructed that consists of a single automated station in which control of level, pressure and / or discharge can be done.

For the control of the new station, a modular PLC of the company B&R Industrial Automation GmbH of the X20 series is used, specifically the X20CP1381, using the Automation Studio software.

The project presents different pressure control programs of the station using the Structured Text programming language. In the programs the GEMMA guide is used and a PID controller is implemented in the pressure regulation program.

ÍNDEX

1. INTRODUCCIÓ.....	10
1.1. MARC TEÒRIC.....	10
1.1.1. Evolució de la indústria.....	10
1.1.2. Evolució de l'automatització en la indústria.....	11
1.1.3. Reguladors PID	15
1.1.4. Sistema SCADA (Supervisor Control And Data Acquisition).....	16
1.1.5. Normativa IEC-61131 i els PLCs de última generació	17
1.1.6. Industria 4.0 i servidors OPC	21
1.2. JUSTIFICACIÓ DEL TREBALL.....	24
1.3. OBJECTIUS DEL TFG.....	24
2. ANTECEDENTS	26
2.1. DESCRIPCIÓ DE LES ESTACIONS DISPONIBLES	26
2.2. TREBALLS PREVIS	31
3. DISSENY DE L'ESTACIÓ DE TREBALL	32
3.1. CONSTRUCCIÓ DE LA MAQUETA.....	32
3.1.1. 1ª Fase: Definició de l'estació i material necessari.	32
3.1.2. 2ª Fase: Distribució dels elements que formen l'estació en la taula de treball i circuit del cablejat fins el quadre elèctric.	35
3.1.3. 3ª Fase : Disseny i distribució del quadre elèctric de l'estació..	37
3.1.4. 4ª Fase: Disseny del cablejat i localització dels senyals en els components del quadre elèctric.	38
3.1.5. 5ª Fase: Identificar els senyals i cablejar.	39
3.2. FUNCIONALITATS DE LA MAQUETA	39
3.3. DESCRIPCIÓ DELS SENSORS I ACTUADORS.....	42
3.3.1. Sensors	42
3.3.2. Actuadors	46

3.4. TECLAT D'OPERACIÓ	49
4. SISTEMA DE CONTROL I AUTOMATITZACIÓ	50
4.1. PLC'S DE LA MAQUETA	50
4.1.1. X20 CP 1583.....	50
4.1.2. X20 CP 1381.....	55
4.2. CONNEXIONS AMB ELS SENSORS I ACTUADORS.....	66
5. ENTORN DE PROGRAMACIÓ	70
5.1. NOU PROJECTE EN AUTOMATION STUDIO.....	70
5.2. CONNEXIÓ AMB EL PLC	77
5.3. INCORPORAR NOU HARDWARE.....	80
5.4. PROGRAMACIÓ AMB TEXT ESTRUCTURAT	80
5.5. PROGRAMACIÓ I SIMULACIÓ.....	86
5.6. AGREGAR UN OBJECTE VISUALITZADOR	87
6. AUTOMATITZACIÓ DE L'ESTACIÓ.....	89
6.1. NIVELLS D'AUTOMATITZACIÓ	89
6.2. AUTOMATITZACIÓ DE L'ESTACIÓ DE TREBALL	89
6.2.1. Recirculació/Neteja del vas d'expansió	89
6.2.2. Monitorització de variables.....	105
6.2.3. Programació de PIDs	108
6.2.4. Sintonia dels paràmetres del regulador PID	110
6.2.5. Automatització completa del sistema de pressió	112
6.3. SISTEMA SCADA	120
7. PROBLEMES I SOLUCIONS.....	123
7.1. PROBLEMES DE PROGRAMA.....	123
7.1. PROBLEMES DE DISSENY EN L'ESTACIÓ.....	123
8. CONCLUSIONS.....	124
8.1. TASQUES PROPOSADES	124

9. BIBLIOGRAFÍA	126
10. ANNEXOS	128
10.1. ANNEX 1: PROGRAMES	128
10.1.1. Variables globals i Cyclic dels programes.....	128
10.1.2. Programa 1.....	129
10.1.3. Programa 2.....	132
10.1.4. Programa 3.....	135
10.1.5. Programa 4.....	138
10.1.6. Programa 5.....	142
10.1.7. Programa 1 amb control PID	153
10.1.8. Programa 2 amb control PID	161
10.2. NORMA DE GRAUS DE PROTECCIÓ IP	171
10.3. IMPLEMENTAR I SINTONITZAR PARÀMETRES D'UN PID AMB BLOC DE FUNCIONS EN AUTOMATION STUDIO	173

ÍNDEX D'IMATGES o FIGURES

Imatge 1 - Control PID en un sistema de llaç tancat.	15
Imatge 2 –Diagrama de Funcions Seqüencials (SFC) (Martín, 2016).	19
Imatge 3 – Llenguatges de programació de la norma IEC-61131 (Martín, 2016)	20
Imatge 4 - Esquema i foto del procés de control de nivell.	27
Imatge 5 - Esquema i foto del procés de control de cabal.	28
Imatge 6 - Esquema i foto del procés de control de pressió.	28
Imatge 7 - Esquema i foto del procés de control de temperatura.	29
Imatge 8 - Senyalització de les electrovàlvules i el motor de comú de l'estació distribuïdora.	30
Imatge 9 - Estació de distribució.	30
Imatge 10 - Estacions de control de nivell i caudal antigues.	32
Imatge 11 - Base per construir l'estació de control de nivell, pressió i cabal.	34
Imatge 12 - Esquema de l'estació 1.	35
Imatge 13 - Distribució aproximada de l'estació.	36
Imatge 14 - Fotografia de l'estació organitzada.	37
Imatge 15 - Distribució del quadre elèctric.	38
Imatge 16 - Circuit de control de nivell i cabal	40
Imatge 17 - Circuit de control de pressió de l'estació 1.	41
Imatge 18 - Sensor de cabal FT1.1.	42
Imatge 19 - Sensor capacitiu de proximitat LT1.1 – LT1.5 i el seu diagrama de funcionament.	43
Imatge 20 - Sensor d'ultrasons LT1.6 i la seva corba característica.	44
Imatge 21 - Aïllador de 2 vies, esquema de funcionament, entrades i sortides.	45

Imatge 22 - Sensor de pressió PT1.1 amb corba característica.	46
Imatge 23 - Bomba hidràulica P1.1.	46
Imatge 24 - Variador de freqüència o regulador del motor i esquema I/O.	47
Imatge 25 - Electrovàlvula o vàlvula solenoide de dues vies V1.1 – V1.4.	48
Imatge 26 - Electrovàlvula analògica o proporcional V1.5.	48
Imatge 27 - Teclat d'operacions.	49
Imatge 28 - Panell d'estat de la CPU del PLC X20CP1583.	51
Imatge 29 - Panell d'estat de l'alimentació del PLC X20CP1583.	53
Imatge 30 - Selector de mode i botó de "Reset" del PLC X20CP1583.	53
Imatge 31 - Mòdul d'alimentació i de comunicació RS232 del PLC X20CP1583.	54
Imatge 32 - PLC X20CP1583.	55
Imatge 33 – Panell del mòdul X1 del PLC X20CP1381.	56
Imatge 34 - Entrades i sortides de les interfícies de comunicació CAN i RS232 del PLC X20CP1581 (mòdul X1).	59
Imatge 35 - Interruptor per habilitar / deshabilitar la resistència de finalització de bus CAN del PLC X20CP1581 (mòdul X1).	59
Imatge 36 - Pinout i exemple del mòdul X1 del PLC X20CP1581.	60
Imatge 37 - Pinout i exemple del mòdul X2 del PLC X20CP1581.	61
Imatge 38 - Panell del mòdul X2 del PLC X20CP1381.	62
Imatge 39 - Pinout i exemple del mòdul X3 del PLC X20CP1581.	63
Imatge 40 - Panell del mòdul X3 del PLC X20CP1381.	64
Imatge 41 - PLC X20CP1381.	64
Imatge 42 - Pinout i exemple del mòdul X4 del PLC X20CP1581.	65
Imatge 43 - Panell del mòdul X4 del PLC X20CP1381.	65
Imatge 44 – Pantalla d'inici per crear un nou projecte en Automation Studio.	70

Imatge 45 - Finestra de creació d'un projecte nou en Automation Studio.	70
Imatge 46 - Finestra de configuració d'un projecte nou en Automation Studio.....	71
Imatge 47 - Finestra d'identificació de hardware manual d'un projecte nou en Automation Studio.....	71
Imatge 48 - Finestra d'identificació automàtica de hardware d'un projecte nou en Automation Studio.	72
Imatge 49 - Entorn de treball d'Automation Studio.....	73
Imatge 50 - Finestra de "Logical, Configuration i Physical View" d'Automation Studio.....	73
Imatge 51 - Inserir un objecte de programació en Automation Studio.....	74
Imatge 52 - Selecció del "Cyclic" del programa en Automation Studio.....	75
Imatge 53 - Crear una variable en Automation Studio.	75
Imatge 54 - Selecció de tipus de variable en Automation Studio.	76
Imatge 55 - Finestra d'ajuda d'Automation Studio.....	77
Imatge 56 - Finestra "Online Settings" amb el PLC trobat (Automation Studio).	77
Imatge 57 - Canviar la IP de la connexió Ethernet del PC.	78
Imatge 58 - Canviar la IP de la connexió Ethernet en Automation Studio.	79
Imatge 59 - Conectar-se al PLC un cop ja configurades les IP en Automation Studio.	79
Imatge 60 - Incorporació del mòdul X4 (X20AO2622) en Automation Studio.....	80
Imatge 61 - Diagrama de temps de la funció TON.....	85
Imatge 62 - Eines per compilar, transferir, monitoritzar el programa, i eines per controlar i simular el PLC en Automation Studio.	86
Imatge 63 - Inserció d'un objecte visualitzador en el projecte en Automation Studio.	87
Imatge 64 - Buscar un visualitzador ja configurat en VNC Viewer.	88
Imatge 65 - Grafcet del programa 1.....	91
Imatge 66 - Grafcet del programa 2.....	92

Imatge 67 - Grafcet del programa 3.	93
Imatge 68 - Grafcet del programa 4.	95
Imatge 69 - Estats de la guia GEMMA.	96
Imatge 70 - Guia GEMMA del programa 5.	98
Imatge 71 - Grafcet de la guia GEMMA del programa 5.....	99
Imatge 72 - Teclat d'operacions segons la guia GEMMA del programa 5.	100
Imatge 73 - Grafcet en el mode automàtic del programa 5.....	101
Imatge 74 - Grafcets individuals dels actuadors del programa 5.	102
Imatge 75 - Grafcet d'aturada d'emergència del programa 5.	103
Imatge 76 - Grafcet de la guia GEMMA amb les transicions definides del programa 5. .	104
Imatge 77 - Ubicació de l'eina "Watch".	105
Imatge 78 - Insertar variables en l'eina "Watch".	105
Imatge 79 - Agregar variables a l'eina "Watch".	106
Imatge 80 - Activar el mode "monitor".	106
Imatge 81 - Forçar les variables des de l'eina "Watch".	106
Imatge 82 - Propietats de l'eina "Trace".	107
Imatge 83 - Estat del "Trace".	107
Imatge 84 - Guardar gràfic en un arxiu de dades.	108
Imatge 85 - Exemple de programa pel controlador PID.....	110
Imatge 86 - Grafcet de rutina de control del programa1_PID.	113
Imatge 87 - Grafcet de rutina d'inici del programa1_PID.	114
Imatge 88 - Grafcet de rutina de funcionament normal del programa1_PID.	115
Imatge 89 - Grafcet de rutina de posta en l'estat inicial del programa1_PID.	116
Imatge 90 - Grafcet de rutina d'emergència del programa2_PID.	117

Imatge 91 - Grafcet de rutina d'inici del programa2_PID.	118
Imatge 92 - Grafcet de rutina de funcionament normal del programa2_PID.	118
Imatge 93 - Grafcet de rutina de control del programa2_PID.....	119
Imatge 94 - Grafcet de rutina de posta en l'estat inicial del programa2_PID.....	119
Imatge 95 - Sistema SCADA: Pantalla inicial.....	120
Imatge 96 - Sistema SCADA: Neteja de vas (programa 5).	121
Imatge 97 - Sistema SCADA: Control PID (programa 2 PID).....	122

ÍNDEX DE TAULES

Taula 1 - Diferències històriques entre DCS i PLC.....	14
Taula 2 - Taula d'estats de la CPU del PLC X20CP1583.....	52
Taula 3 - Taula d'estats del sistema de comunicacions "POWERLINK".....	52
Taula 4 - Taula d'estats del sistema d'alimentació del PLC X20CP1583.....	53
Taula 5 - Taula de modes d'operació del PLC X20CP1583.	54
Taula 6 - Taula d'estats de la CPU del PLC X20CP1581.....	57
Taula 7 - Taula d'estats de la comunicació POWERLINK del PLC X20CP1581.....	58
Taula 8 - Possibles funcions de les entrades digitals de comptatge ràpid DI11 a DI14 del mòdul X2 del PLC X20CP1581.....	61
Taula 9 - Estats del mòdul X4 del PLC X20CP1381.....	66
Taula 10 – Taula de connexions aïllador de 2 vies.	66
Taula 11 - Taula de connexions Variador.....	67
Taula 12 - Taula de connexions terminal analògic.	67
Taula 13 - Taula de connexions Mòdul X1.	68
Taula 14 - Taula de connexions mòdul X2.	68
Taula 15 - Taula de connexions del mòdul X3.	68
Taula 16 - Taula de connexions del mòdul X4.	69
Taula 17 - Taula de paraules clau en Text Estructurat.....	81
Taula 18 - Taula d'operadors en Text Estructurat.	82
Taula 19 - Taula de parametres del temporitzador TON (ST).	85
Taula 20 - Resultats de les variables de control PID.....	109
Taula 21 - Resultats de la sintonia de paràmetres de control PID.....	111

1. INTRODUCCIÓ

El present projecte es basa en fer una maqueta funcional aprofitant el material de quatre estacions antigues i incorporant-hi dos nous PLC de *B&R Industrial Automation GmbH* de la sèrie X20. En el projecte es vol conèixer les bases de la indústria 4.0, i veure com s'apliquen aquestes bases, així també com conèixer les normatives relacionades amb els PLC (IEC 61131) i familiaritzar-se amb un entorn de control utilitzant el software de programació de "Automation Studio de B&R".

Durant el transcurs del projecte es pretén adquirir més coneixements pràctics sobre automatització i control industrial així també com desenvolupar més habilitats d'instal·lació de components electrònics i habilitats de programació amb text estructurat.

1.1. Marc teòric

1.1.1. Evolució de la indústria

La indústria ha estat crucial per l'evolució de la nostra societat gràcies a la gran població que ocupa en zones desenvolupades, el que requeria una gran producció de béns, com ara la indústria tèxtil o la indústria del sector metal·lúrgic. Aquest augment de la producció demanava reduir el temps de producció i mecanitzar-la de manera que es milloressin els temps de procés amb la reducció de l'esforç físic per a la producció de béns en massa (Cervantes).

La indústria 1.0 també coneguda com a primera revolució industrial, va començar al segle XVIII a Gran Bretanya i es va estendre per Europa entre l'any 1820 i el 1840, i va durar fins a mitjans del segle XIX, va ser el principi de la indústria tèxtil així també com avanços en la tecnologia i les tècniques agrícoles que permetia un augment de la productivitat. També es coneix com l'era de la revolució en els transports a causa de l'aparició de la màquina de vapor.

La coneguda com a indústria 2.0 o també anomenada com segona revolució industrial va de l'any 1850 fins al 1870. Es coneix com una època on el sector de la indústria va cobrar força degut a les migracions cap a la ciutat de la gent que vivia en els pobles. Es caracteritza sobretot pel descobriment de l'electricitat que va substituir, com a nova font d'energia, la màquina de vapor. En aquesta època hi va haver grans investigadors en el camp de l'electricitat (i derivats) en destaquem Nikola Tesla en camp de l'electromagnetisme, Thomas Alva Edison i George Westinghouse en el camp de l'electricitat i l'indústria. Es va

començar a fer servir els motors de combustió interna com a conseqüència del desenvolupament i la investigació en el camp del petroli, i també es van crear les primeres línies de muntatge (Henry Ford).

La tercera revolució industrial o indústria 3.0 no està del tot definida per grans successos en la història però aproximadament apareix en els 70. Es poden apreciar grans avenços en la tecnologia de la indústria i l'automatització gràcies a la introducció d'electrònica més desenvolupada, així com la millora de les màquines de control numèric i de l'automatització de processos productius. També es parla de nous conceptes com l'energia 100% renovable, l'emmagatzematge d'energia i del vehicle elèctric, entre altres. El terme més important va ser el de les *tecnologies smart grid* o xarxes elèctriques intel·ligents, caracteritzada per la integració dinàmica dels desenvolupaments en enginyeria elèctrica, l'emmagatzement energètic i els avenços de les tecnologies de la informació i la comunicació (TIC).

Actualment ens trobem en el que s'anomena la indústria 4.0 o quarta revolució industrial en la qual apareix l'anomenada indústria intel·ligent. En aquest marc es proposa una nova manera d'organitzar els mitjans de producció amb les anomenades *smart factories* (fàbriques intel·ligents) amb una àmplia interconnexió entre màquines automatitzades, xarxes de comunicacions, integració de tecnologies avançades de processaments de dades, robòtica avançada, capacitat d'autodiagnòstic de situacions, millor intercanvi d'informació i una major eficiència en la gestió de recursos naturals i humans. D'aquesta manera, es busca la flexibilització i personalització del procés productiu de manera que s'adapti de forma intel·ligent a situacions fortuïtes, permetent portar una producció a gran escala, però de forma controlada, sense generar massa estoc.

Les noves tecnologies aplicades en el camp de la indústria sempre han estat exponent de precisió, qualitat i rendiment entre altres. És important doncs conèixer els nous elements que es fan servir en la indústria per incrementar tant en temps de resposta en processos industrials com en increment de velocitat en adquirir dades a temps real i tractar-les per obtenir la millor resposta possible el més ràpid possible.

1.1.2. Evolució de l'automatització en la indústria

En l'automatització de processos industrials s'han estudiat les tecnologies aplicades en sistemes o dispositius per unificar el sistema de producció. Abans d'executar qualsevol projecte industrial és convenient fer un estudi dels elements de control necessaris segons les característiques que envolten la industrialització que es portarà a terme. Entre altres cal tenir en compte: les dimensions, la complexitat, el suport del proveïdor o integrador del sistema, el manteniment a llarg o curt termini i els costos del cicle de vida dels components. Segons la indústria que es vol realitzar es requeriran uns elements de control o uns altres (Javier Román). En aquest estat de l'art revisem dues estructures de control els sistemes de control distribuïts i els controladors lògics programables.

1.1.2.1. Sistema de control distribuït

Un sistema de control distribuït (DCS: Distributed Control Systems) permet realitzar el control de seccions individuals d'una planta emprant controladors de procés locals. Aquest sistema consta d'estacions d'operació que actuen com a interfícies home-maquina, controladors on resideix la lògica, mòduls d'entrada i sortida que permeten la connexió a perifèrics i a xarxes de comunicació que les interconnecten. Normalment els mòduls I/O es troben distribuïts per la planta en estacions remotes per reduir el cablejat, de la mateixa manera estan connectats a controladors per protocols de comunicació de camp (bus de camp o *fieldbus*).

Una característica dels DCS és la redundància, disposen de varis: controladors, mòduls I/O, servidors i xarxes de comunicació, fet que incrementa la tolerància a les falles (continuitat del procés en cas d'avaries d'un dels components), incrementa per tant la disponibilitat de l'equipament..

En l'estructura dels DCS hi ha tres nivells bàsics:

- **Nivell d'operació:** Aquest nivell és el d'interacció del sistema amb els operadors de la planta i és on es troben els sistemes informàtics per a la monitorització del procés i l'adquisició d'informació en temps real, la qual s'emmagatzema en una base de dades per posteriorment analitzar-les i generar històrics. Aquest nivell gestiona a més l'intercanvi d'informació amb el sistema de manteniment i el sistema de planificació de la producció.
- **Nivell de control:** En aquest nivell es troben els controladors de les diferents parts del procés. Aquests controladors estan interconnectats entre si i entre les estacions d'operació mitjançant xarxes de comunicació.
- **Nivell de camp:** En aquest nivell hi trobem els perifèrics que controlen les diferents variables així com altres I/O que tenen la funció de control de procés. Aquests perifèrics han de ser compatibles amb els sistemes de bus de camp disponibles: Profibus, Fieldbus i protocol HART entre altres.

En 1974, l'empresa Honeywell va ser la primera companyia que va llançar el primer Sistema de Control Distribuït anomenat TDC 2000 (les sigles provenen de l'angles: *Total Distributed Control*). Va ser dissenyat principalment pel control de processos de grans indústries (entre 3.000 i 20.000 punts de I/O); en contraposició amb el control d'esdeveniments discrets de màquines de manufactura realitzat normalment amb PLCs. Inicialment es va utilitzar per substituir elements analògics amb interfícies que simulaven la vista frontal d'aquests panells ("faceplates"). La majoria dels senyals que es monitoritzaven eren analògics (de 4 a 20 mA, de 0 a 10 V, etc), tals com la temperatura, pressió, caudal, pH, etc.

DCS destaca per la funcionalitat pel control continu (llaços de control PIDs), control seqüencial, control per lots (batch), algorismes de control complexos, entre altres.

Un dels grans canvis en els últims 20 anys ha estat reemplaçar hardware propietari per computadors personals (PCs) i tecnologies de xarxa d'àrea local (LAN: Local Àrea Network) estàndard (Ethernet, TCP/IP). Això a comportat una reducció dels costos i simplificació d'instal·lació. Amb el temps s'ha familiaritzat amb la norma IEC 61131, possibilitant la seva instal·lació en processos o plantes petites (< 1.000 I/O).

El canvi més important d'alguns DCS és el fet d'incorporar el concepte de sistemes de control oberts (OCS), facilitant la integració de productes de tercers mitjançant estàndards industrials basats en OLE/COM (Object Linking and Embedding / Common Object Model) com ara OPC (O/e per control de processos).

1.1.2.2. Controladors lògics programables

Un PLC (Controlador Lògic Programable) és un controlador electrònic capaç d'executar una lògica programada per software per control industrial de màquines, línies d'assemblatge, etc.

Un PLC es compon d'una unitat de processament (CPU), mòduls d'entrades i sortides (I/O) i interfícies de comunicació. Es caracteritza per ser escalable tant en potència de càlcul com en quantitat d'entrades i sortides, així també com versàtil ja que pot treballar connectat a una xarxa o bé treballar aïllat. El PLC és fàcil de programar i disposa de diversos llenguatges de programació regits per la norma IEC 61131-3. En cas de necessitar supervisió del procés, es connecten interfícies homes-màquina (HMIs) que ja poden ser de panells industrials a PCs (Personal Computers) amb un software SCADA (control, supervisió i adquisició de dades) que compleix les funcions d'adquisició i operació de procés.

Es va dissenyar a finals dels '60 per substituir la lògica de relés utilitzada en control discret que feien servir sistemes de lògica combinacional. En 1968 Bedford Associates va fer el primer PLC anomenat 084 (pel fet que el nom provenia del número de projecte), un producte que el van anomenar com Modicon (Modular Digital Controller). La programació inicial va ser en "lògica d'escala" (ladder) que simula el cablejat elèctric utilitzat amb relés, així es pretenia tractar amb aplicacions en processos on la majoria dels senyals d'entrada i sortida eren digitals i es requerien altes velocitats d'execució (temps de cicle d'ordre de mili segons).

En 1971, un equip d'enginyers de l'empresa Allen-Bradley, va començar a desenvolupar un nou concepte de controlador industrial que incorporava millores basades en les necessitats dels clients, creant d'aquesta manera el 1774 PLC.

Les habilitats de comunicació van començar a aparèixer en 1973. El primer sistema va ser el bus Modicon (Modbus). El PLC podia ara dialogar amb altres PLC's, en conjunt podien estar aïllats de les màquines que controlaven. També podien enviar i rebre senyals de tensió variables, entrant en el món analògic. Desafortunadament, la manca d'un estàndard acompanyat amb un continu canvi tecnològic va fer que la comunicació de PLC'S fos un gran oceà de sistemes físics i protocols incompatibles entre si.

En els anys vuitanta els components electrònics permetien executar operacions de 16 bits, a diferència dels de 4 bits que es feia servir als 70s. Durant aquesta època es va produir un intent d'estandardització de les comunicacions amb el protocol MAP (Manufacturing Automation Protocol) de General Motors. També va ser un temps en el qual es van reduir les dimensions del PLC i es va passar a programar amb programació simbòlica a través d'ordinadors personals en comptes dels clàssics terminals de programació.

A començament dels noranta, van aparèixer els microprocessadors de 32 bits amb possibilitat d'operacions matemàtiques complexes, i de comunicacions entre PLCs de diferents marques i PC, aquest fet va obrir la possibilitat de realitzar fàbriques completament automatitzades amb comunicació amb la Gerència en "temps real".

1.1.2.3. Comparativa entre DCS i PLCs

Les diferències entre aquests sistemes es mostren en la següent taula.

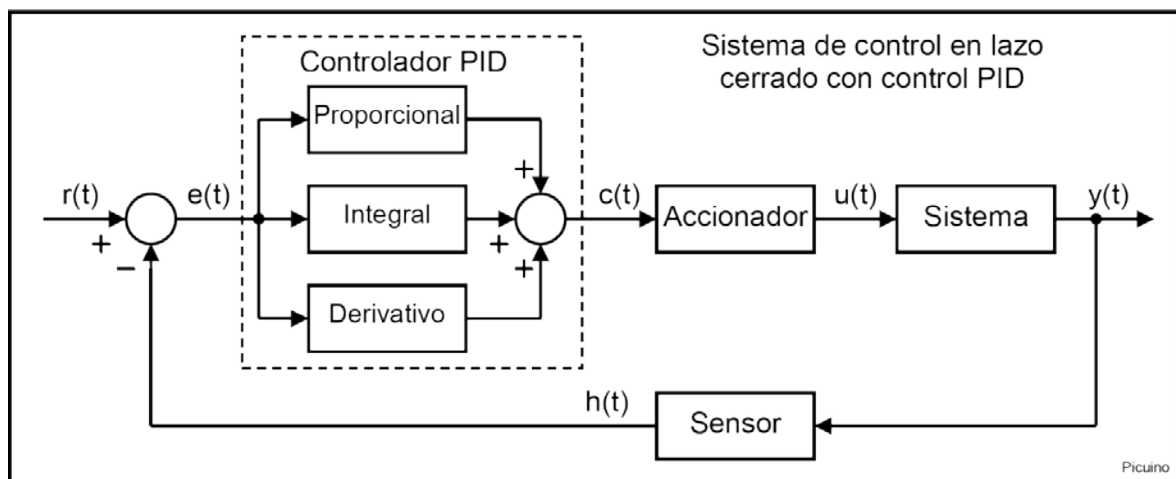
Taula 1 - Diferències històriques entre DCS i PLC.

Criteri	DCS	PLC
Indústries	De Procés	Màquines
Aplicació	Control Continu	Control Discret
Majoria de I/Os	Analògics	Digitals
Temps de Cicle	Dècimes de segon	Mil·lèsimes de segon
Dimensions típiques	2.000 a 20.000 I/Os	10 a 2.000 I/Os
HMI	Interna	Externa
Base de dades	Única	Diverses

Actualment, la línia que separa els mons dels PLCs i DCS no està clara ja que alguns DCS es canvien per PLCs que gestionen processos. Això és pel fet que el hardware dels dispositius de control són similars (mateixos processadors), deixant les diferències entre PLC i DCS a nivell de software. Actualment també existeix un mercat per sistemes híbrids que requereix funcionalitats tant de control de procés i com de control d'esdeveniments discrets en els quals els PLCs i els DCS tenen el seu rol. En aquests casos cadascú aporta els seus avantatges en l'àrea de procés tot potenciant l'acció dels operaris amb la capacitat de prendre les millors decisions amb supervisió global.

1.1.3. Reguladors PID

En els processos industrials es requereix adaptar els senyals de sortida per obtenir una resposta controlada i adequada als senyals adquirits pels sensors (analògic o digital) que llegeix variables físiques del sistema. El controlador PID es basa en un control amb realimentació (llaç tancat) que permet corregir l'error del sistema. Perquè el controlador PID pugui regular un procés necessita un sensor que recapti les dades actuals del sistema, un controlador que generi el senyal que controla l'actuador i un actuador que modifiqui l'estat del sistema. El controlador té tres accions de control que permeten adequar el senyal de sortida segons l'error que tingui sistema (imatge 1):



Imatge 1 - Control PID en un sistema de llaç tancat.

- **Acció de control proporcional (P):** La part proporcional consisteix en el producte entre la constant proporcional i l'error, és a dir, dona una sortida que es proporcional a l'error. Aquesta acció permet aproximar l'error en estat estacionari a zero encara que si aquesta constant és massa gran pot causar sobre oscil·lacions desestabilitzant el sistema, per raons de seguretat no pot sobrepassar del 30%.

$$y(t) = Kp \cdot e(t)$$

- **Acció de control integral (I):** La part integral dona una sortida que és proporcional a l'error acumulat (desviació permanent de la variable pel que fa al punt de consigna) de manera que disminueix l'error en estat estacionari provocat per pertorbacions externes. L'error es corregirà en el temps, a diferència del proporcional, ja que el fet que hi hagi una integral ens dona la mitjana de l'error d'un període determinat de temps.

$$y(t) = Ki \cdot \int_0^t e(\tau) d\tau$$

Acció de control derivativa (D): La part derivativa actua només quan el valor absolut de l'error varia (quan l'error no és constant) per tant l'error és la desviació actual entre el valor de mesura i la consigna. Aquest control permet corregir l'error proporcionalment amb la mateixa velocitat que es genera, evitant que s'incrementi l'error.

$$y(t) = Kd \cdot \frac{de}{dt}$$

El control PID consta de la suma de les tres accions, podent corregir l'error del sistema eficaçment i en el mínim temps possible:

$$y(t) = Kp \cdot e(t) + Ki \cdot \int_0^t e(\tau) d\tau + Kd \cdot \frac{de}{dt}$$

Algunes d'aquestes accions de control no són necessàries, ja que alguns sistemes no tenen un comportament gaire lineal o la variable mesurada presenta molt soroll, en aquests casos es poden utilitzar altres estructures de control amb controladors que apliquin les accions necessàries (P, P+I, P+D) per garantir una estabilitat en el sistema.

1.1.4. Sistema SCADA (Supervisor Control And Data Adquisition)

A la indústria actual cada cop és més important l'ús d'equips que tinguin incorporat un sistema de control i supervisió dels dispositius de les màquines tant a nivell remot com a nivell de camp, tant en apartats de fabricació com a nivell de distribució o emmagatzematge (MOLLO, 2014).

El desenvolupament dels microprocessadors, microcontroladors i dels controladors lògics programables han fet que pugui controlar una o varies variables del sistema realitzant un control directe sobre les mateixes, el que es coneix com a control distribuït. Aquests equips de control "local" es comuniquen amb dispositius del seu mateix nivell i amb altres de nivell superior de supervisió.

L'estructura de control estàndard que es fa servir en la indústria està dividida en dues parts:

- Part operativa: Aquí es troben els actuadors i la part del control visual del procés que dona informació a l'operari de l'estat de la màquina així també com la informació necessària per dur a terme les operacions necessàries a planta.
- Part de control: En aquesta part es pot trobar els dispositius de control (PLCs, DCPs i PCs industrials) que gestionen els senyals als actuadors basant-se amb les accions assignades per software i els senyals dels sensors rebuts des de planta. Per altra banda també hi trobem els elements de comunicació que van des del controlador cap a altres elements de gestió o control.

Per dur a terme el control visual es fa servir el software SCADA (Supervisor Control And Data Acquisition) un sistema d'adquisició de dades i control de supervisió capaç d'adaptar-se a les necessitats de control d'un procés. Aquest software està dissenyat per treballar sobre ordinadors en el control de la producció, proporcionant comunicació amb els dispositius de camp (controladors autònoms, autòmats programables, etc.) de manera que es pot controlar i supervisar processos industrials aportant informació a temps real. Els elements de procés del sistema de control poden utilitzar una implementació estàndard en temps real per la comunicació entre objectes a través de xarxes. La base de la comunicació normalment es fa servir amb xarxes d'àrea local o bé fent servir altres bussos especials.

Bàsicament un sistema SCADA es denomina als programes necessaris i hardware addicional. Les característiques d'un sistema SCADA són generar panells d'alarma que permeten a l'operari reconèixer els problemes que té la màquina en cada instant, així també com recaptar dades històriques del procés que s'està controlant. En termes de control, un sistema SCADA permet modificar les lleis o tasques de control, com ara bé la modificació de paràmetres de la màquina dins d'uns valors de seguretat predefinits. L'espai de treball d'alguns sistemes SCADA ofereixen llibreries de funcions per llenguatges d'ús general que permeten personalitzar amb detall l'aplicació que es vol realitzar.

1.1.5. Normativa IEC-61131 i els PLCs de última generació

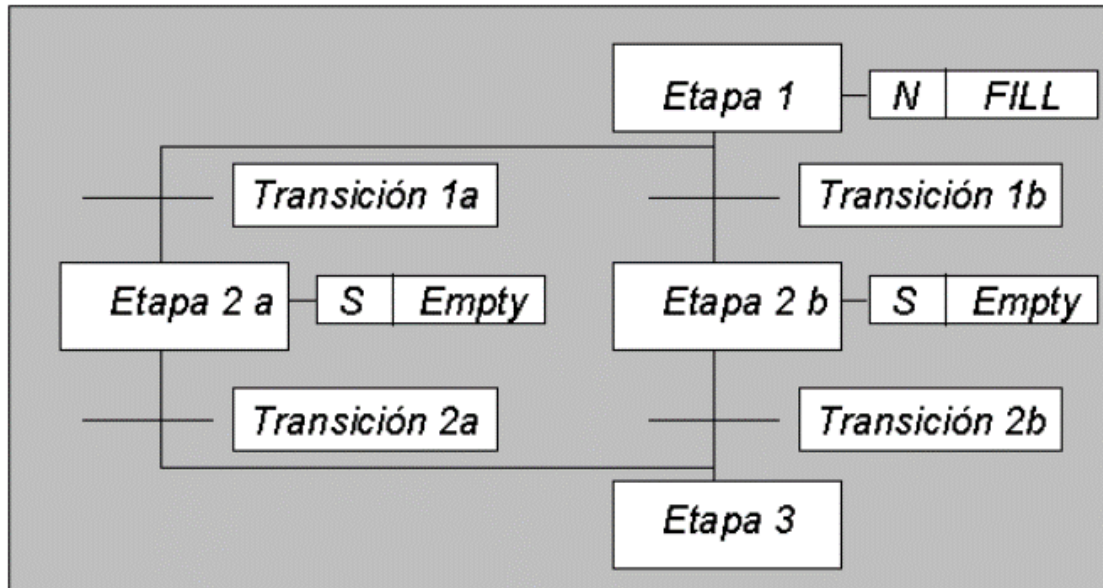
La normativa IEC-61131 és un estàndard de la *International Electrotechnical Commission*, o en català Comissió Electrotècnica Internacional, que defineix les especificacions dels sistemes basats en Controladors Lògics Programables (PLC) tant en hardware com en software pel desenvolupament d'algoritmes genèrics pels usuaris de processos industrials (Pineda Sánchez). La norma especifica els requisits mínims per les característiques funcionals així també com les condicions de servei i la seguretat general aplicables als PLCs i als seus perifèrics. Es defineix tant els llenguatges de programació més utilitzats com les normes sintàctiques i semàntiques que es tenen de respectar a més de les comunicacions entre els PLCs i altres sistemes.

La IEC-61131 està definida en vuit parts (PLCopen, s.f.):

- IEC-61131-1 (Informació general): Conté definicions que es fan servir en aquesta norma com ara equip de comunicació de dades (DCE), equip terminal de dades (DTE), sistema de control lògic, processador principal (MPU), interfície home-màquina (MMI), temps mig entre fallades (MTBF), equip de programació i posta a punt (PADT), estació d'entrada / sortida remota (RIOS), equip d'assaig (TE), entre altres. També conté característiques funcionals que distingeixen els sistemes basats en PLC d'altres sistemes:
 - Funció de tractament del senyal.
 - Funció d'interfície amb els sensors i actuadors.
 - Funció de comunicació.
 - Funció d'interfície home-màquina.
 - Funcions de programació, posada a punt, assaig i documentació.
 - Funcions d'alimentació de corrent.
- IEC-61131-2 (Especificacions i proves dels equips): En aquest apartat s'hi exposa els requisits elèctrics (alimentació de corrent alterna (AC) o contínua (DC), E/S analògiques o digitals, processadors principals i memòria del sistema, etc), mecànics (magneto tèrmics, cables i connectors, requisits de la bateria, disposicions per la terra de protecció, etc) i funcionals pels autòmats programables i els perifèrics corresponents així com les condicions de servei, emmagatzematge i transport aplicables. Aquest apartat també incorpora les proves de qualitat que tenen de complir, les condicions ambientals i els mètodes i procediments d'assaig que es tenen d'utilitzar (verificació de les característiques de l'alimentació de AC i DC, verificació de les característiques d'entrada / sortida, verificació de les característiques del processador principal, verificació de les estacions d'E/S remotes, verificació de les característiques dels perifèrics, verificació de l'autodiagnòstic i diagnòstic).
- IEC-61131-3 (Llenguatges de programació): En aquest part es descriu la sintaxi i semàntica de cinc llenguatges de programació per PLC. En aquesta part es fa una descripció dels elements comuns com ara el tipus de dades, variables, configuració recursos i tasques així també com unitats d'organització de programa. Alhora d'estructurar els programes, en la norma es proposa l'ocupació d'unitats de programació anomenades "POU's" (*Programming Organization Units* - Unitats Organitzatives de Programa) que permeten fer grans programes amb un conjunt de programes que desenvolupen una funció específica, similar als blocs funcionals (FB – *Functional Block*) que ofereixen de sèrie els PLC en el seu software de programació.

Per descriure un programa es fa mitjançant Diagrama de Funcions Seqüencials (SFC) que descriu el comportament seqüencial del programa de control. Permet dividir el problema en parts facilitant el diagnòstic de problemes i tasques de manteniment. És un derivat de les xarxes de Petri i Grafset, ja que consisteix en un conjunt d'etapes en les

quals es realitza una sèrie d'accions i unes transicions que permeten l'evolució d'una etapa a la següent. A la Imatge 2 es mostra un exemple d'un diagrama de funcions.

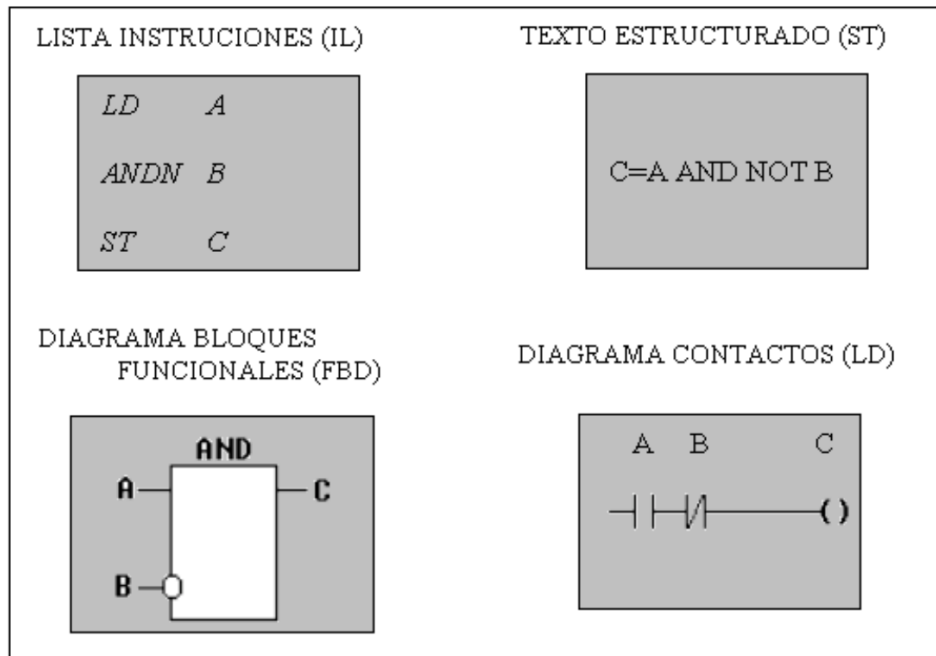


Imatge 2 –Diagrama de Funcions Seqüencials (SFC) (Martín, 2016).

La selecció del llenguatge depèn del programador, l'aplicació concreta, del nivell de definició de l'aplicació, de l'estructura del sistema de control i del grau de comunicació:

- Diagrama Escala (LD): Està basat en la representació gràfica de la lògica de contactes. Simula les connexions dels relés que es troben en la instal·lació.
- Llista d'Instruccions: Es un llenguatge de baix nivell que es similar al muntatge de la instal·lació. El control del programa (flux de control) s'aconsegueix mitjançant instruccions de salt i trucades de funció (subrutines amb paràmetres opcionals).
- Text Estructurat: És un Llenguatge d'alt nivell amb arrels en Ada, Pascal y C. Conté tots els elements essencials d'un llenguatge de programació modern, inclosa la selecció de flux d'execució (IF, THEN, ELSE, CASE OF...) i llaços d'iteració (FOR, WHILE...). Aquest llenguatge és ideal per la definició de blocs de funcions complexes que poden ser utilitzats en altres programes.
- Diagrama de Blocs de Funcions (FBD): Normalment es fa servir en la indústria de procés. Representa el comportament del programa mitjançant un conjunt de blocs de funcions equivalents a diagrames de circuits d'electrònica.

En la imatge 3 es mostra un petit exemple de com funcionen els llenguatges esmentats anteriorment.



Imatge 3 – Llenguatges de programació de la norma IEC-61131 (Martín, 2016) .

- IEC-61131-4 (Guia de l'usuari (TR – Technical Report)): Té com a finalitat ajudar als usuaris finals a seleccionar i especificar els requisits dels seus equips d'acord amb el IEC61131.
- IEC-61131-5 (Comunicacions): Descriu els serveis de comunicació des del punt de vista del programador i / o usuari. Com a tal, és una interfície de programa d'aplicacions per a la comunicació PLC. Per a això proporciona serveis de comunicació en forma de funcions combinades amb els conceptes i elements dels idiomes de programació IEC 61131-3. Aquesta norma permet que el Client i el Servidor es puguin trobar i entendre's, o canviar el rol de client i servidor.
- IEC-61131-6 (Seguretat funcional): Aquesta part de la sèrie IEC 61131 especifica els requisits per als controladors programables (PLC) i perifèrics associats, tal com es defineixen a la part 1, que es pretenen utilitzar com a subsistema lògic d'un sistema relacionat amb la seguretat elèctric/electrònic/electrònic programable (E/E /PE). Un controlador programable i els seus perifèrics associats que compleixin els requisits d'aquesta part es consideren adequats per a ser utilitzats en un sistema relacionat amb la seguretat E / E / PE i s'identifica com un controlador lògic programable de seguretat funcional (FS-PLC). Un FS-PLC és generalment un subsistema de maquinari (HW) /

programari (SW). Un FS-PLC també pot incloure elements de programa, per exemple blocs de funció predefinits.

- IEC-61131-7 (Programació de control difús): El control difús s'està generant com una tecnologia que pot millorar les capacitats d'automatització industrial, i és adequat per a tasques de nivell de control generalment realitzades en controladors programables (PC). El control difús es basa en el coneixement de l'aplicació pràctica representada per les anomenades bases de regles lingüístiques, en lloc de per models analítics (ja siguin empírics o teòrics). L'aplicació del control difús pot ser avantatjós en aquells casos en què no hi hagi un model de procés explícit disponible o en què el model analític és massa difícil d'avaluar o quan el model és massa complicat per avaluar-lo en temps real, ja que permet tenir el coneixement disponible per millorar els processos i realitzar diverses tasques, per exemple:
 - Control de sistemes amb circuit tancat o obert, únic o multi-variable, per a sistemes lineals o no lineals.
 - Configuració en línia o fora de línia dels paràmetres dels sistemes de control.
 - Classificació i reconeixement de patrons.
 - Presa de decisions en temps real (envieu aquest producte a la màquina A o B?) ajudant els operadors a prendre decisions o definir els paràmetres.
 - Detecció i diagnòstic de fallades en sistemes.
- IEC-61131-8 (Guies de programació (TR – Technical Report)): Proporciona als usuaris de sistemes de control programable tal com es defineixen a IEC 61131-3, com ha de programar, configurar, instal·lar i mantenir controladors programables que formen part de sistemes de mesura i control de processos industrials. També dona pautes per a la implementació dels llenguatges en sistemes de control programables i els seus entorns de suport a la programació (PSE).

1.1.6. Indústria 4.0 i servidors OPC

L'anomenada indústria 4.0 o quarta revolució industrial apareix de l'anomenat internet de les coses (IoT – *Internet of things*) on la incorporació de tecnologies de la informació (TIC - Tecnologies de la informació i comunicacions) i altres avenços tecnològics en la indústria fan de les fabriques actual un model innovador anomenat "Smart Factory" o "Internet industrial". Aquesta nova indústria on la robòtica i la sensòrica prenen un paper important en la gestió de la producció dona un pas més en donar nous serveis gracies a les noves tecnologies de software, connectivitat i capacitat de control simultani de cada estació de treball (Mazaeda).

La quarta revolució industrial és similar a la tercera revolució però potenciada per la revolució digital. La digitalització de la indústria permet a l'empresa incorporar un manteniment

predictiu eficaç, millor presa de decisions a temps real, anticipar els inventaris basats en la producció (el subministrament de material i les comandes es fan automàticament), millor coordinació entre treballs, entre altres sistemes de gestió de la indústria, permetent d'aquesta manera personalitzar la producció segons les necessitats del client. El fet de digitalitzar la indústria i estendre el flux de dades mitjançant xarxes d'alta velocitat i de gran capacitat permet una comunicació instantània eliminant els límits de la planta industrial individual i tenint control de varies plantes industrials des de qualsevol punt on es tingui un accés al servidor de l'empresa.

Els termes més arrelat a la indústria 4.0 (Navarro., 2018):

- **Internet de las cosas (IoT):** Els elements que registren el procés de producció han evolucionat de manera que incorporen sistemes de comunicació que permeten interconnectar-se entre ells millorant la presa de decisions i actuant de manera més ràpida. Actualment es fan servir transductors que incorporen dispositius d'identificació per radio freqüència (RFID) per informar a les estacions de treball del procés a dur a terme.
- **Sistemes ciber-físics (CPS):** Els sistemes ciber-físics estan constituïts per tecnologies TIC i sistemes de processament encastats, encara que antigament els sistemes encastats feien referència al procés. Ara aquests sistemes incorporen sistemes de comunicació basats amb IoT i constitueixen sistemes distribuïts intel·ligents que evolucionen de forma autònoma.
- **Big Data:** Com a conseqüència de la gran quantitat de dades (Volum), el número d'operacions a ser executades (Velocitat) i la quantitat de dades (varietat) que apareixen en el concepte de Smart Factory, apareix aquest concepte de Big Data on cal utilitzar mètodes avançats per extreure patrons repetitius dins dels blocs de dades per aplicar un model predictiu de recaptació de dades. Aquestes dades s'avaluen amb l'objectiu de fer-ne un ús comú de dades entre els equips de producció, els proveïdors i els clients per facilitar la presa de decisions i poder identificar patrons en el producte que puguin generar productes defectuosos.
- **Cloud Computing:** Els serveis de tecnologia de la informació (TI) cada cop es reinventen més per resoldre les carències tecnològiques com la complexitat creixent d'administrar tota la infraestructura de dades i software on aplicar una solució factible es tradueix en costos elevats. Cloud Computing aplica una solució més eficient fent servir núvols privats i/o públics per desenvolupar una plataforma, infraestructura o software que ofereixi un servei de gestió i/o emmagatzematge de dades.
- **Simulació:** En un futur pròxim es preveu incorporar un sistema que mitjançant l'adquisició de dades dels sistemes de control de planta es pugui generar un model virtual que permeti simular procediments de producció que augmentin el rendiment i eficiència de

la planta. Posteriorment aquestes simulacions s'aplicaran a la planta en temps real i permetrà ajustar les màquines de manera òptima i reduir els temps de parada alhora que s'augmenta la qualitat. Actualment les simulacions ja s'apliquen a dissenys de nous productes però es pretén aplicar a planta.

- **Robòtica:** En els últims anys la robòtica s'ha aplicat en processos repetitius, perillosos o complexos però la tecnologia actual redissenya els robots perquè aquests puguin ser autònoms i col·laboratius. Els robots autònoms tenen com a peculiaritat que no necessiten cap intervenció humana, ja que poden reconèixer i aprendre del seu entorn i prendre decisions per ells mateixos, el robot AGV (vehicle de guiatge automàtic) és un exemple. Els robots col·laboratius o cobots permet fer tasques que a l'operari li assumeix gran esforç físic o un cert grau de perillositat, mantenint sempre una seguretat als xocs amb l'operari, ja que el robot es para en detectar una variació en els valors establerts de força (variant segons el treball que ha de fer).
- **Visió Artificial:** És una visió des d'un ordinador d'un cert procés o element. Normalment s'aplica en qualitat o bé en simulació de moviment d'un robot amb l'ajuda d'una càmera que adquireix dades, processa la imatge i l'analitza. Amb aquests sistemes es pretén aplicar:
 - Inspecció visual automàtica que permeti analitzar un producte.
 - Identificar peces amb la finalitat de detectar-les o classificar-les segons cada procés.
 - Guiar un robot industrial amb l'ajuda de sensòrica per determinar la localització exacta d'una peça o per millorar el posicionament en alguns dels seus moviments.
- **Realitat augmentada:** És un tipus d'entorn de visualització interactiu que permet millorar la informació adquirida per un sistema que presenti tecnologia IoT. Amb aquesta tecnologia podem adquirir informació rellevant sobre els elements de la planta com ara bé instruccions, manuals, lloc on s'ubiquen les peces de recanvi, etc, així també com tenir una visió clara d'on es troben els problemes d'una màquina i informació de com ha d'actuar l'especialista o l'operari.
- **Fabricació Additiva (FA):** En l'actualitat, les impressores 3D han agafat molta força en molts sectors, ja que permeten crear peces de recanvi, implants, joguines, etc. Aquestes impressores es basen en l'aplicació de resina i plàstic o altres materials implantant capes, l'una sobre l'altre fins a haver complert el programa enviat (normalment 3D CAD) a la impressora. En el camp de la indústria s'està fent servir per adaptar-se a les demandes dels clients personalitzant la producció. D'altra banda també es fa servir per imprimir prototips que es volen incorporar en un futur en un procés o en la producció, i així poder analitzar la factibilitat del model; reduir els costos de reparació considerablement. Millorar en la qualitat.

En l'actualitat es fa servir un protocol de comunicació anomenat OPC (*O*/*le* per control de processos), basat en un sistema client-servidor. El sistema OPC és un estàndard que estableix un idioma comú per l'intercanvi d'informació entre diferents dispositius (sistemes SCADA, entre altres) y fabricants de manera que dona flexibilitat per recórrer a diferents proveïdors mantenint la connectivitat entre ells. Aquest sistema soluciona el problema de quan una aplicació requeria l'accés a un element de control, era necessari una interfície específica o un controlador per al diàleg entre l'aplicació i l'element de control.

Els servidors OPC normalment fan servir ethernet per a la comunicació a planta i també fan servir un núvol per emmagatzemar dades o accedir-hi mitjançant internet. Els servidors són segurs gràcies a què fan servir un sistema encriptat de dades quan es transfereix informació.

1.2. Justificació del treball

Aquest treball neix de la necessitat de familiaritzar-se amb un sistema automatitzat i adaptar-se amb els nous elements de control que presenta l'indústria actual. El fet de disposar d'una estació de control de nivell i pressió, amb un PLC d'última generació i una àmplia manera de programar-lo, permet entendre de primera mà el funcionament de la gestió d'un procés industrial amb un sistema d'adquisició com el que ens ofereix els PLC de B&R. També cal considerar que aquest PLC està especialment dissenyat per adaptar-se a l'indústria 4.0 i en conseqüència, incorpora altres software útil per a les indústries intel·ligents.

1.3. Objectius del TFG

El treball fi de grau desenvolupat té per objectiu el desenvolupament d'una maqueta o estació de treball útil per a treballar les tecnologies de la indústria 4.0. Més concretament el treball realitzat consisteix en el disseny una estació que permet simular un sistema híbrid en el qual intervenen variables lògiques (digitals) i variables contínues (analògiques), i automatitzar-les mitjançant un autòmat programable industrial (PLC) de última generació. L'estació resultant s'hi podrà realitzar un control de nivell, pressió i cabal.

L'objectiu principal s'ha estructurat en els següents objectius específics:

- Entendre el concepte d'automatització d'un procés incorporant noves tecnologies de control que estan involucrades en aquesta quarta revolució industrial, és a dir, conèixer més profundament els elements de control, des del muntatge i instal·lació del nou sistema de distribució i diferents elements que el componen fins a la seva programació des del PLC.
- Realitzar una maqueta totalment funcional.

- Crear un entorn físic de control i un entorn HMI (Interfície Home-Maquina) pel control de les estacions.
- Dissenyar una sèrie de programes didàctics per entendre el control de l'estació i familiaritzar-se amb l'entorn de "B&R Industrial Automation GmbH" i les diferents maneres de programar els seus PLC.

2. ANTECEDENTS

El projecte ha estat producte de reciclar estacions antigues que es feien servir per a l'aprenentatge de l'automatització i control a nivell industrial de la companyia FESTO, fent servir PIDs per a la regulació de diferents variables de procés i un PLC per a gestionar els diferents sensors i actuadors que es podien trobar en la maqueta. La maqueta constava de 4 estacions sobre rodes que permetien fer la regulació de diferents variables de procés, en concret nivell, caudal, pressió i temperatura, i a part una estació de distribució que permet connectar els dipòsits principals de cada estació.

La idea principal és dissenyar una estació que es pugui unificar els controls de nivell, de caudal i de pressió, sent així molt més dinàmica i més compacta. En aquest cas es volia automatitzar l'estació mitjançant dispositius d'última generació que incorporessin noves tecnologies pel que es refereix a gestió de senyals i velocitat de processat.

Una empresa coneguda com a *B&R Industrial Automation GmbH* ha subministrat a la universitat (EPSEM) uns nous PLC els quals eren més nous que el de les estacions subministrades per FESTO (OMRON CP1L) i incorporaven varis llenguatges de programació i hardware HMI (pantalla tàctil) que permet programar-lo amb sistema SCADA.

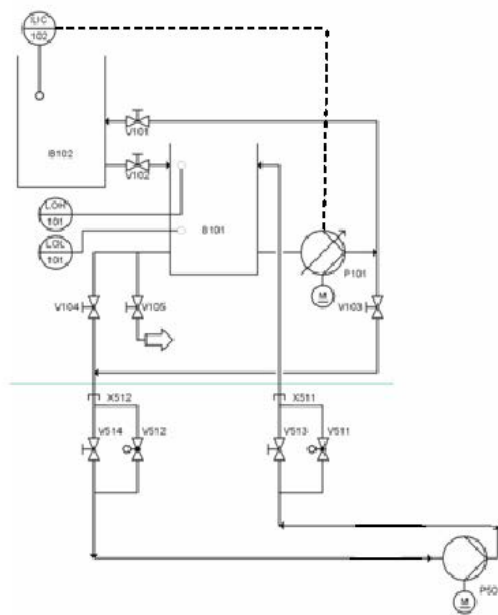
2.1. Descripció de les estacions disponibles

1. Estació de control de nivell

Aquesta estació es basa en el control de nivell mitjançant dos tancs interconnectats amb tubs de PVC, vàlvules manuals i una motobomba controlada mitjançant un PID. L'estació també incorpora dos sensors capacitius de nivell en tanc inferior i un sensor analògic en tanc superior. La motobomba es fa servir per recircular l'aigua del dipòsit inferior al superior. L'estació compta amb clau de pas per la sortida de líquid pel dipòsit inferior per si es vol extreure aigua del sistema. En la imatge 4 es pot veure l'esquema i una fotografia de l'estació. La part inferior de la imatge on s'observen electrovàlvules i una bomba formen part de l'estació distribuïdora.

La gestió de l'actuador (bomba), el sensor analògic d'ultrasons i els digitals es fa mitjançant un PID industrial. El terminal analògic 170699 es on connecten tots els sensors, actuadors i elements de control de les estacions antigues. El mateix terminal té marcat on cal connectar els transductors (Pt100, Level, Flow, IN BIN o Binary Input, 24V, 10V, GND, etc) . El terminal incorpora un connector SYSLINK pel controlador de llaç tancat de cada estació.

L'estació compta amb una botonera per portar el control de l'estació així també com amb una petita pantalla de control amb una botonera bàsica. La botonera es basa en 4 interruptors: on/off del procés (POWER), on/off de la bomba (PUMP), on/off de la vàlvula motoritzada (VALVE), un interruptor sense acció (EXTERNAL FUNCTION) i 2 indicadors de control per visualitzar que el procés està engegat (llum verda) i un altre d'alarma (llum blava). El control de l'estació només és possible mitjançant variacions en la tensió d'alimentació de la bomba.

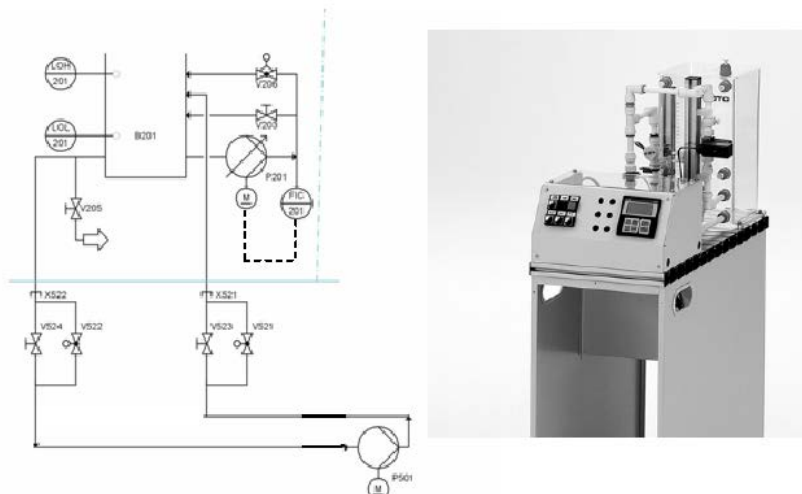


Imatge 4 - Esquema i foto del procés de control de nivell.

2. Estació de control de cabal

Aquesta estació té com a objectiu fer el control de cabal. En aquest cas es disposa d'un sol dipòsit d'aigua. En l'estació es pot regular el flux d'aigua circulant fent servir una motobomba o bé amb una electrovàlvula analògica controlada mitjançant un PID industrial. En la imatge 5 es pot veure l'esquema i una fotografia de l'estació. La part inferior de la imatge on s'observen electrovàlvules i una bomba formen part de l'estació distribuïdora.

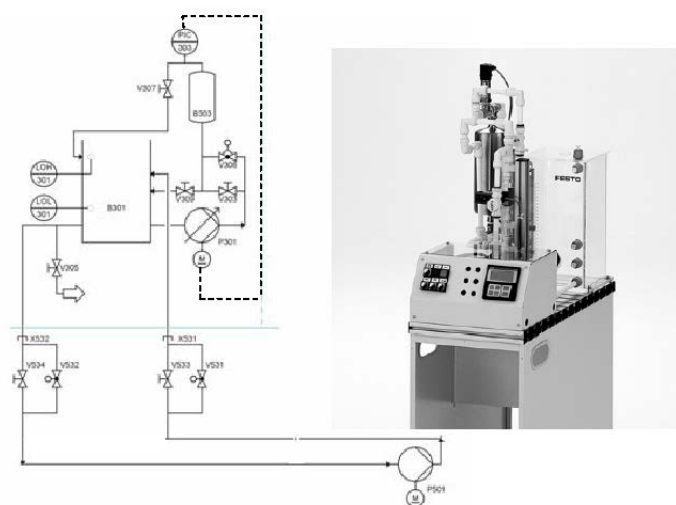
L'estació està equipada amb dos sensors capacitius que ens marquen els límits de nivell del tanc així també com amb una clau de pas per buidar el dipòsit manualment. Aquesta estació també incorpora un teclat d'operacions.



Imatge 5 - Esquema i foto del procés de control de cabal

3. Estació de control de pressió

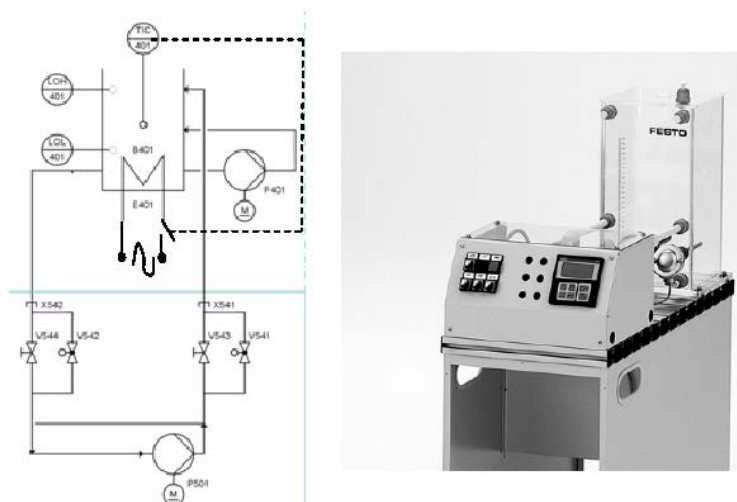
Aquesta estació incorpora un sol dipòsit i un vas d'expansió. Disposa d'un sensor de pressió amb el qual mesurar la pressió que hi ha en el vas d'expansió. Per regular la pressió a l'interior del vas s'utilitza una motobomba. Diferents vàlvules manuals permeten derivar l'aigua al vas o directament al dipòsit. En la imatge 6 es pot veure l'esquema i una fotografia de l'estació. Igual que les altres estacions, incorpora dos sensors capacitius que ens marquen els límits de nivell del tanc així també com amb una clau de pas per buidar el dipòsit manualment. Aquesta estació incorpora un control PID i teclat d'operacions.



Imatge 6 - Esquema i foto del procés de control de pressió.

4. Estació de control de temperatura

L'estació de temperatura consta principalment d'una unitat calefactora i un sensor de temperatura que es troben incorporats en l'únic dipòsit de l'estació. En la distribució de components trobem que hi ha una bomba únicament per recircular l'aigua del dipòsit amb la finalitat de refredar l'aigua més ràpid. A la imatge 7 es pot veure l'esquema i una fotografia de l'estació. Igual que les altres estacions, incorpora dos sensors capacitius que ens marquen els límits de nivell del tanc. Aquesta estació incorpora un control PID i teclat d'operacions.

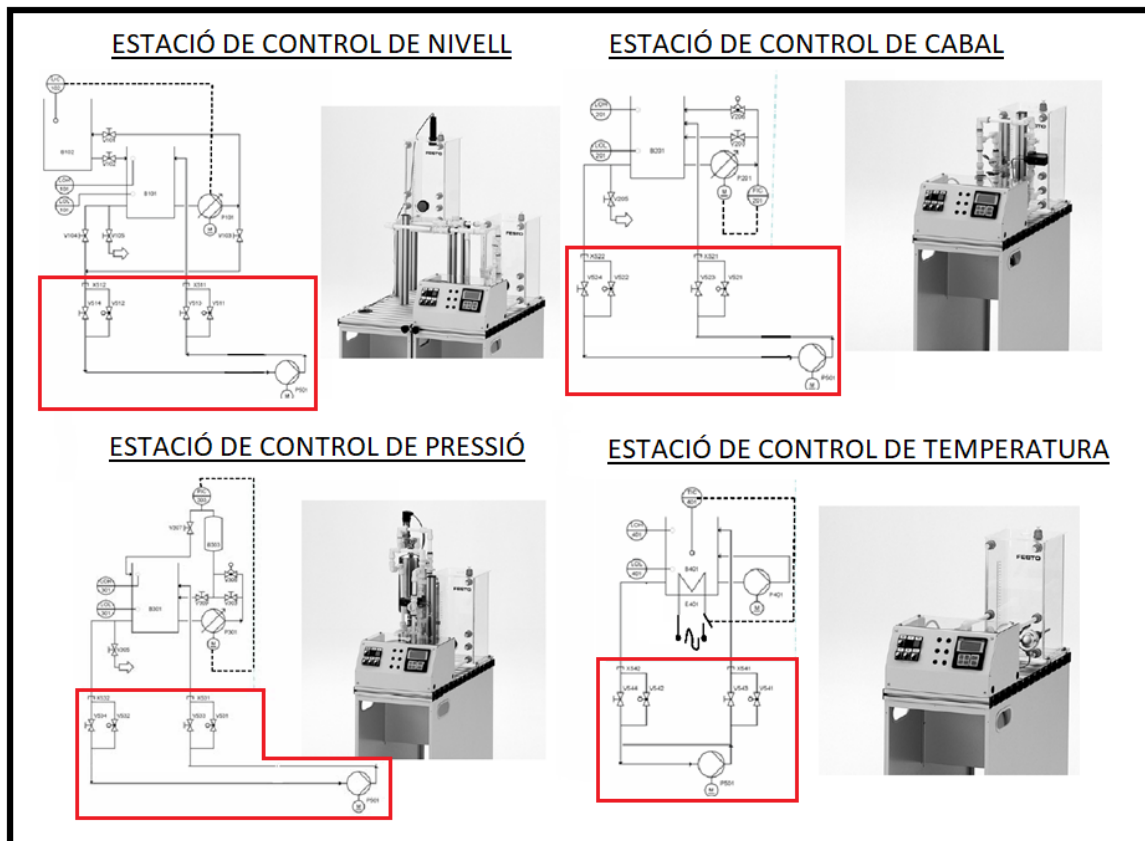


Imatge 7 - Esquema i foto del procés de control de temperatura.

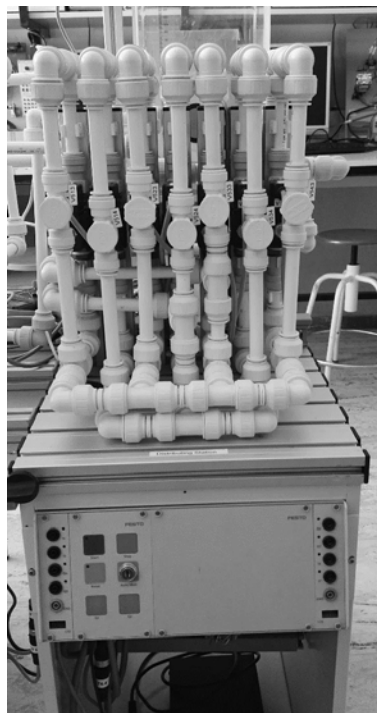
5. Estació distribuïdora

La distribució d'aigua d'una estació a una altra es fa mitjançant l'estació distribuïdora. Els elements que trobem a l'estació són una bomba, vuit electrovàlvules i vuit vàlvules manuals connectades en paral·lel amb les electrovàlvules podent controlar manualment o automàticament el pas de l'aigua. En aquesta estació es té accés als sensors capacitius de les altres estacions, podent moure aigua d'una estació a l'altra. En la imatge 8 es pot veure les parts dels esquemes de les altres estacions que formen part de l'estació distribuïdora i en la imatge 9 es pot veure com és l'estació. El control d'aquesta estació es fa mitjançant un PLC.

La botonera en aquest cas es diferent, incorpora un botó de "start", un de "stop", de "reset" i un selector amb clau del tipus de control (automàtic o manual), també porta dos llums programables. El propi teclat d'operacions incorpora vuit connexions banana femella per recaptar dades de les senyals de l'estació així com dos terres i dos connectors mascle de 5x2 pins per fer-ho servir per entrades o sortides de dades.



Imatge 8 - Senyalització de les electrovàlvules i el motor de comú de l'estació distribuïdora.



Imatge 9 - Estació de distribució.

Els elements que es reciclaran d'aquestes estacions per realitzar el present projecte són els següents:

- 1) Electrovàlvules digitals i electrovàlvula proporcional (analògica).
- 2) Tubs, Colzes i "T" que incorporen les estacions.
- 3) Bombes hidràuliques.
- 4) Carros i suports d'alumini.
- 5) Terminal analògic 170699 on es connecten tots els sensors, actuadors i elements de control de les estacions antigues.

2.2. Treballs previs

Un treball fi de grau realitzat al curs 2017-2018 (Servitja, 2018), va tenir per objectiu el control de les estacions descrites a la Secció 2.1. En el treball va estudiar sobre diferents tècniques de modelització de sistemes i sintonia d'estructures de control PID, arribant a implementar un control PID en el sistema de control de nivell i el sistema de control de pressió.

En el treball referent s'hi aplica diversos controladors PID estudiant el comportament dels sensors de l'estació de control de nivell i pressió segons la tensió aplicada a la bomba hidràulica obtenint les variables necessàries per obtenir un control PID correcte.

Per aquest projecte s'obtindrà dades sobre la sintonia d'un controlador PID mitjançant l'anterior projecte realitzat. S'avaluaran i es tractaran els resultats obtinguts en la present estació de treball.

3. DISSENY DE L'ESTACIÓ DE TREBALL

3.1. Construcció de la maqueta

El projecte partia de les estacions antigues connectades on es va portar a terme un procés de cinc fases per a construir l'estació:

- 1ª Fase: Definició de l'estació (dimensions i elements a incorporar) i material necessari.
- 2ª Fase: Distribució dels elements que formen l'estació en la taula de treball.
- 3ª Fase : Disseny i distribució del quadre elèctric de l'estació.
- 4ª Fase: Disseny del cablejat i localització dels senyals en els components del quadre elèctric.
- 5ª Fase: Identificar els senyals i cablejar.

3.1.1. 1ª Fase: Definició de l'estació i material necessari.

Inicialment teníem les quatre estacions antigues juntes i connectades entre elles (imatge 10). La primera acció va ser separar-les de l'estació distribuïdora.



Imatge 10 - Estacions de control de nivell i caudal antigues.

Pel desmuntatge i la reconstrucció de les estacions es requeria de les següents eines:

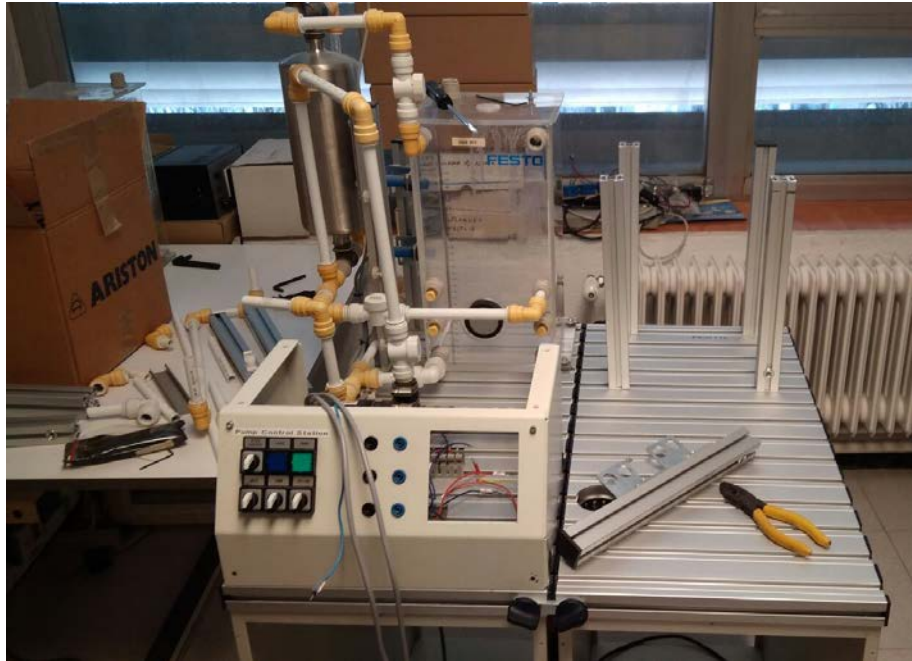
- Talla tubs de PVC.
- Arrugador de punteres buides per a cables elèctrics.
- Multímetre.
- Tisores.
- Pelacables.
- Tornavís de punta plana per components electrònics.
- Tornavís d'estrella.
- Alicates de tall.
- Alicates universals.
- Joc de claus Allen.

També es necessitaven dels següents materials:

- 4 Rotllos de cable elèctric normal negre, marró, groc i verd de 1,5mm de diàmetre.
- 2 Rotllos de cable elèctric normal negre i vermell de 2mm de diàmetre.
- Tubs de PVC de 15mm de diàmetre.
- Colzes i "T" de plàstic amb connectors ràpids de 15mm de diàmetre.
- Brides de plàstic.
- Suports de plàstic per a cables.
- Perfils d'alumini 30x30mm amb ranura "T".
- Terminals de punta per a cables elèctrics.
- Terminals Faston femella per a cables de 1 a 2,5mm de secció.
- Canaleta ranurada per a quadres elèctrics.
- Plaques d'identificació KS amb funda d'identificació KT estàndard.
- Suports metàl·lics diversos per subjectar tubs i electrovàlvules.
- Regleta de connexió de 4mm.

Un cop es disposava del material necessari per a dur a muntar la nova estació, es va plantejar com desenvolupar la nova estació que incorporaria els tres tipus de control (control de nivell, control de pressió i control de cabal). Primer es van considerar els elements que s'havien d'incorporar a l'estació: vas d'expansió, un dipòsit a la base de la taula, un dipòsit superior muntat sobre els perfils d'alumini, la bomba hidràulica i el teclat d'operació de l'estació. Un cop revisat els elements a incorporar i la mida de cadascú, es decideix fer el muntatge utilitzant dues taules.

Es van agafar les tres estacions i es van desmuntar procurant que hi haguessin dues taules per a dissenyar la nova estació. Seguidament es van desmuntar tots aquells elements que inicialment no eren d'interès per a la maqueta, tal com es pot veure en la imatge 11.



Imatge 11 - Base per construir l'estació de control de nivell, pressió i cabal.

Per donar més funcionalitats a la maqueta es va decidir incorporar electrovàlvules en paral·lel amb vàlvules manuals, per tal de poder automatitzar la circulació de l'aigua en els diferents dipòsits.

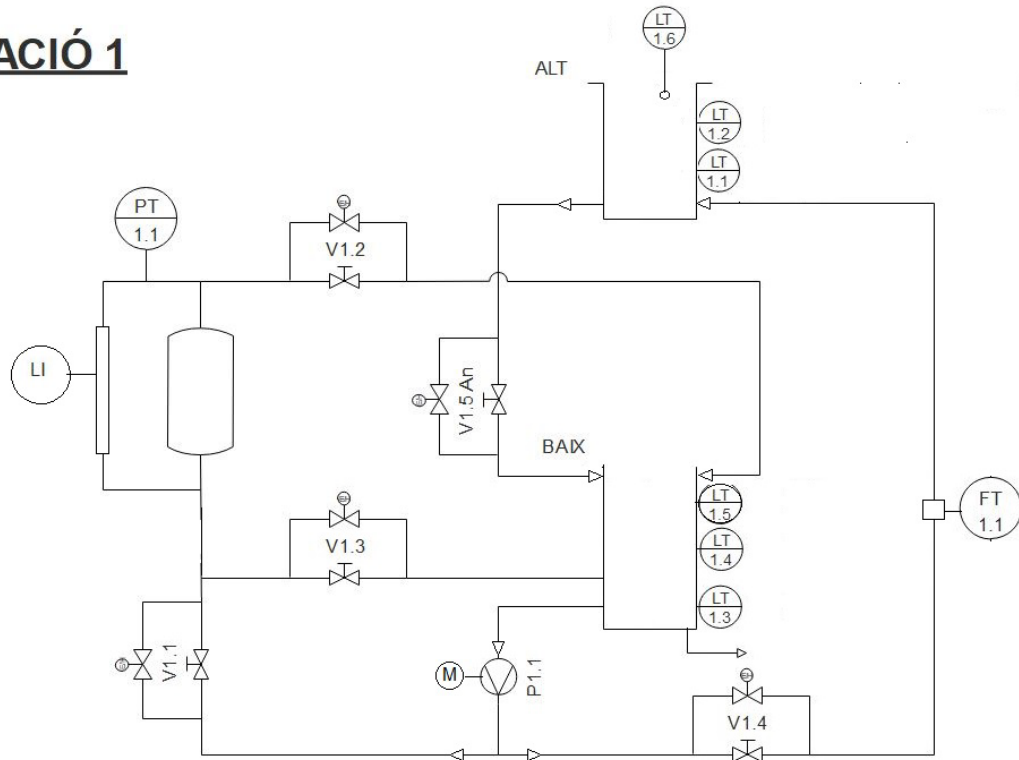
Per l'enumeració dels elements que incorporarem es seguirà l'estructura següent: incorporar simbologia i sigles genèriques segons els tipus de procés que du terme l'element segons la norma DIN 19227, i enumerar l'element primer amb el número de l'estació seguit d'un punt i l'enumeració que correspon segons el número d'elements iguals que hi hagi dins de l'estació.

Considerant els punts esmentats, els elements necessaris per a muntar la maqueta i l'esquema resultant de l'estació 1 (imatge 12) serien els següents:

- Sensors:
 - 5 sensors capacitius de nivell (LT1.1 – LT1.5).
 - 1 sensor d'ultrasons (LT 1.6).
 - 1 Sensor de cabal (FT1.1).
 - 1 Sensor de pressió (PT1.1).
- Actuadors:
 - 4 electrovàlvules digitals (V1.3 – V1.4).
 - 1 electrovàlvula proporcional / analògica (V1.5An).
 - 1 bomba hidràulica (P1.1).
- Altres elements:

- 2 dipòsits de metacrilat.
- 1 Vas d'expansió amb visor (LI).
- 6 vàlvules manuals.

ESTACIÓ 1



Imatge 12 - Esquema de l'estació 1.

Un cop definit els elements que formaran l'estació, es defineixen les ubicacions de cada element dins de la taula.

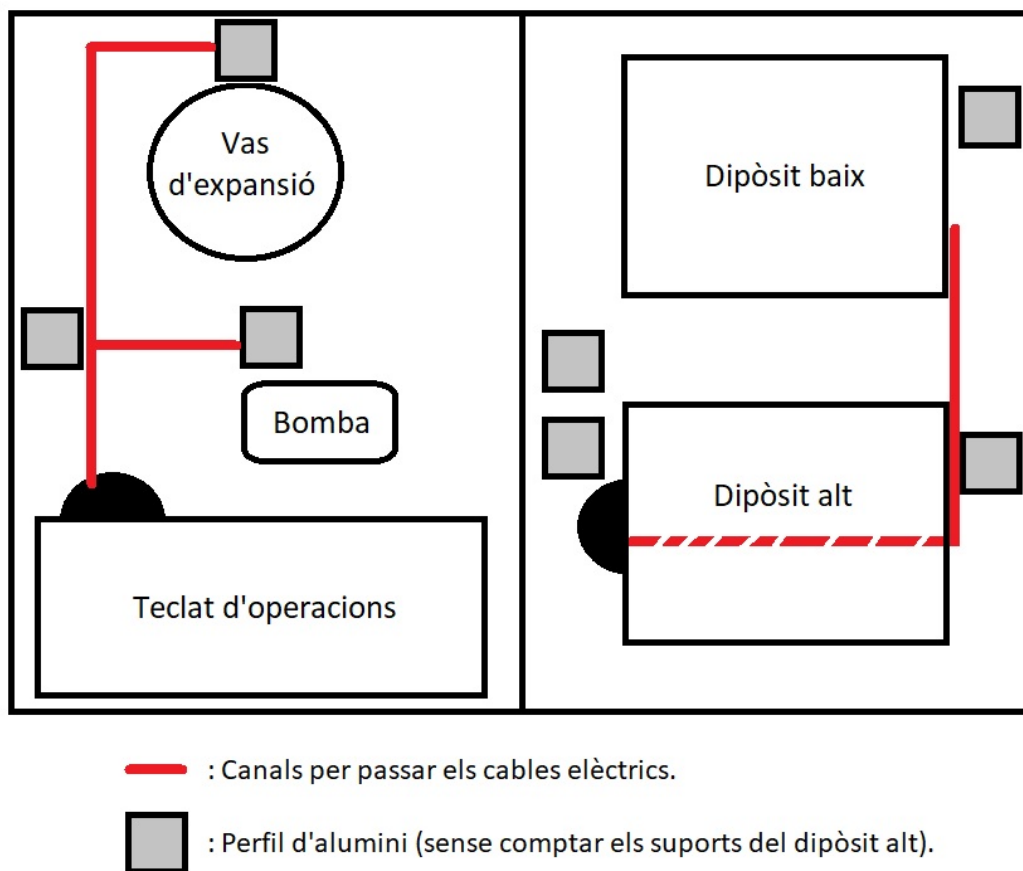
3.1.2. 2ª Fase: Distribució dels elements que formen l'estació en la taula de treball i circuit del cablejat fins el quadre elèctric.

Tant la ubicació del teclat d'operacions com la bomba ja es trobaven en el millor lloc possible, ja que entre la bomba i el teclat, a la banda esquerra de la taula, hi ha un forat per on passar els cables fins al quadre elèctric, i també es la millor ubicació ja que el teclat d'operacions té molts cables per portar els senyals I/O al quadre elèctric.

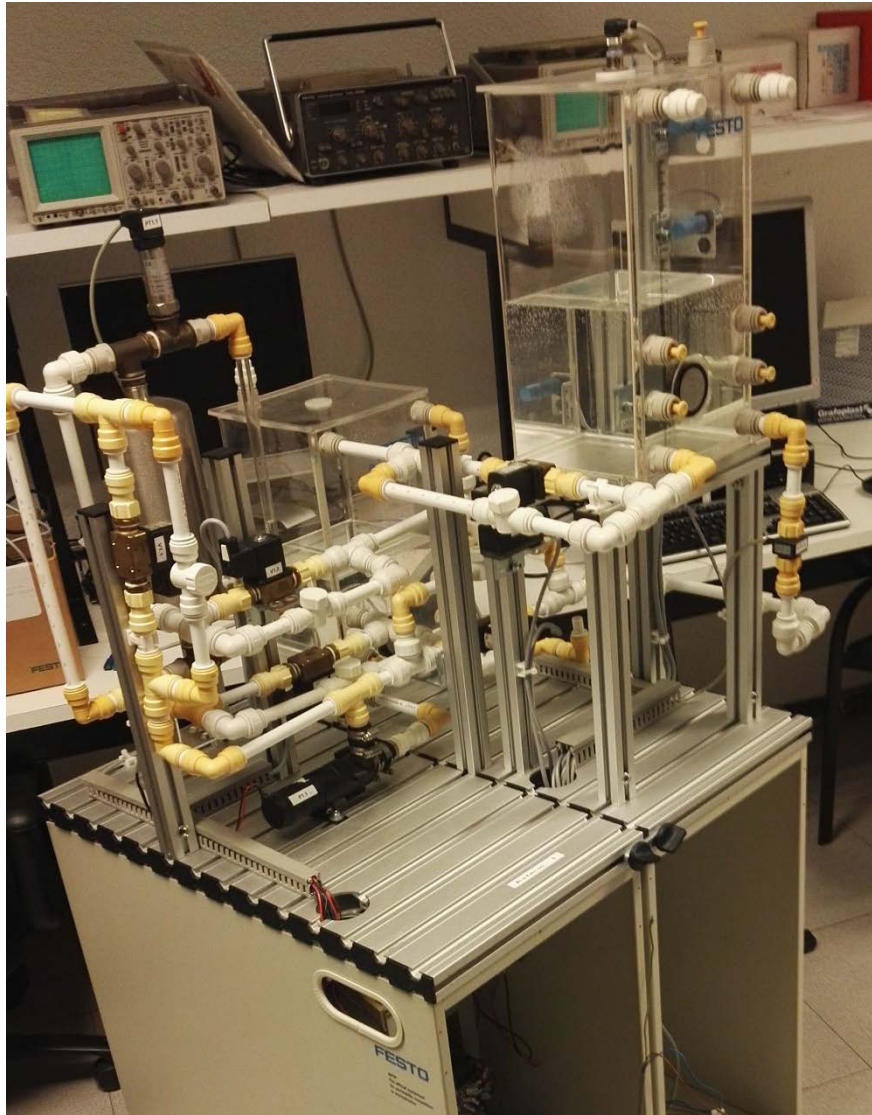
Els altres elements, per temes d'espai es van col·locar en els següents espais de la taula des del punt de vista del teclat d'operacions: el vas d'expansió, en la part posterior de la taula esquerra amb el visor de nivell al costat dret del vas (per facilitar la visualització), just

enrere de la bomba; a la taula de la dreta els dos dipòsits, en la part posterior el dipòsit de baix i en la part frontal el dipòsit superior amb una estructura de perfils d'alumini que incrementés l'altura del dipòsit quedant per sobre del dipòsit de baix. Amb aquesta distribució es podia muntar el sistema d'electrovàlvules i vàlvules manuals en paral·lel a les entrades i sortides de cabal d'aigua. En algunes parts de l'estació es van col·locar perfils d'alumini junt amb suports metàl·lics per subjectar les electrovàlvules i evitar les convexitats en el circuit per culpa del seu pes.

Després de fer diferents proves la distribució va quedar de la forma mostrada a la imatge 13 i 14 dona una idea aproximada de la distribució i del circuit dels cables per l'estació.



Imatge 13 - Distribució aproximada de l'estació.

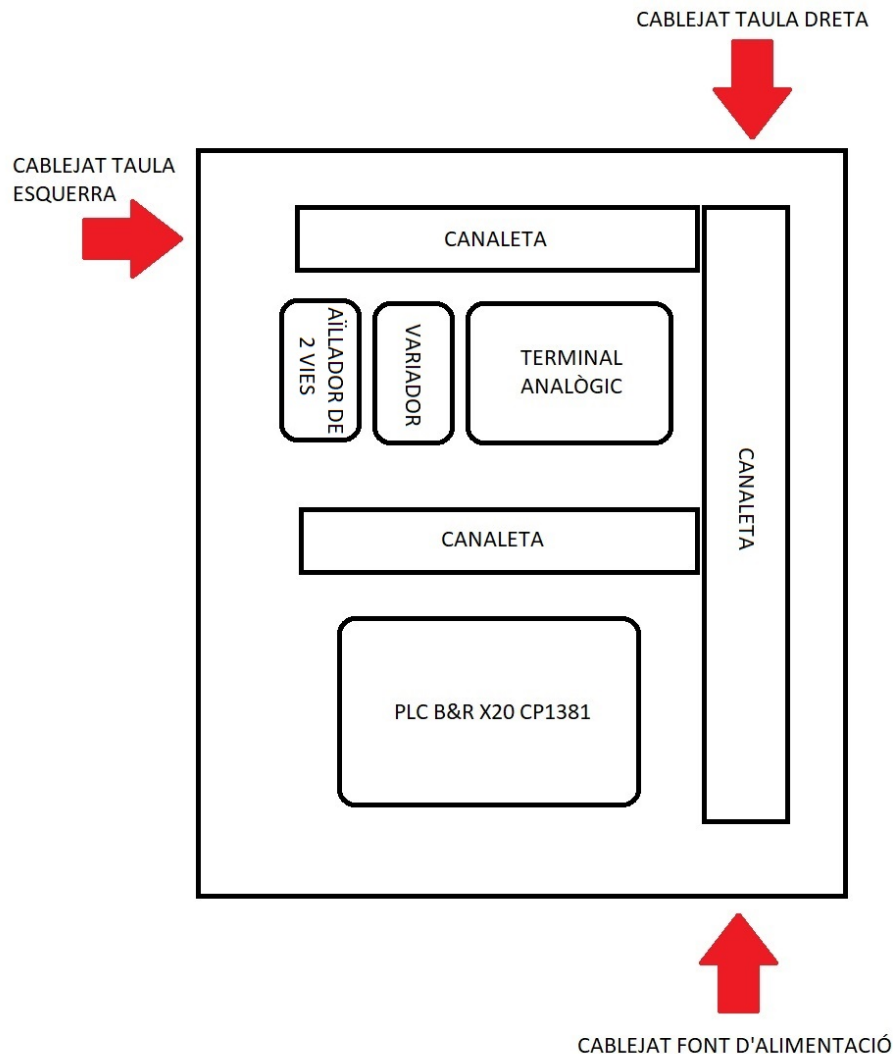


Imatge 14 - Fotografia de l'estació organitzada.

3.1.3. 3ªFase : Disseny i distribució del quadre elèctric de l'estació..

El quadre elèctric consta d'un variador de freqüència pel control de la bomba, un aïllador universal de dues vies per transformar, un terminal analògic on col·locar els senyals alimentats a 24V i a 10V, i els GND comuns, i el PLC modular de B&R X20 CP 1381 amb un mòdul extra per senyals de sortida analògics. Tots aquests elements s'han muntat sobre una plataforma metàl·lica que s'ha ubicat sota la taula, fàcilment extraïble. En la base de la part inferior s'incorporarà una font d'alimentació de FESTO de 24V en contínua amb una intensitat màxima de 4,5A que anirà a alimentar tant el PLC com el terminal analògic. El terra de la font d'alimentació es collarà al suport metàl·lic del quadre elèctric.

La imatge 14 mostra com ha quedat la distribució.



Imatge 15 - Distribució del quadre elèctric

3.1.4. 4ª Fase: Disseny del cablejat i localització dels senyals en els components del quadre elèctric.

En el cablejat, alguns dels cables dels sensors i actuadors van caldre allargar-los. Aquests es van allargar fent servir una regleta de connexions i sempre col·locant una funda KT amb la seva placa d'identificació en una banda i l'altra de la regleta per saber a quin element corresponia en cas que s'afluïxi la regleta i els cables es desconnectessin.

Per norma vam considerar que els cables de color marró ens indicarien generalment alimentació de 24V i vermell o negre generalment senyal de sortida dels transductors, menys

en alguns senyals analògiques o especials que anirien d'un color gris, groc o blanc (per exemple el senyal "teach input" del sensor d'ultrasons de color blanc). El cablejat de terra generalment es troba de color groc/verd.

3.1.5. 5ª Fase: Identificar els senyals i cablejar.

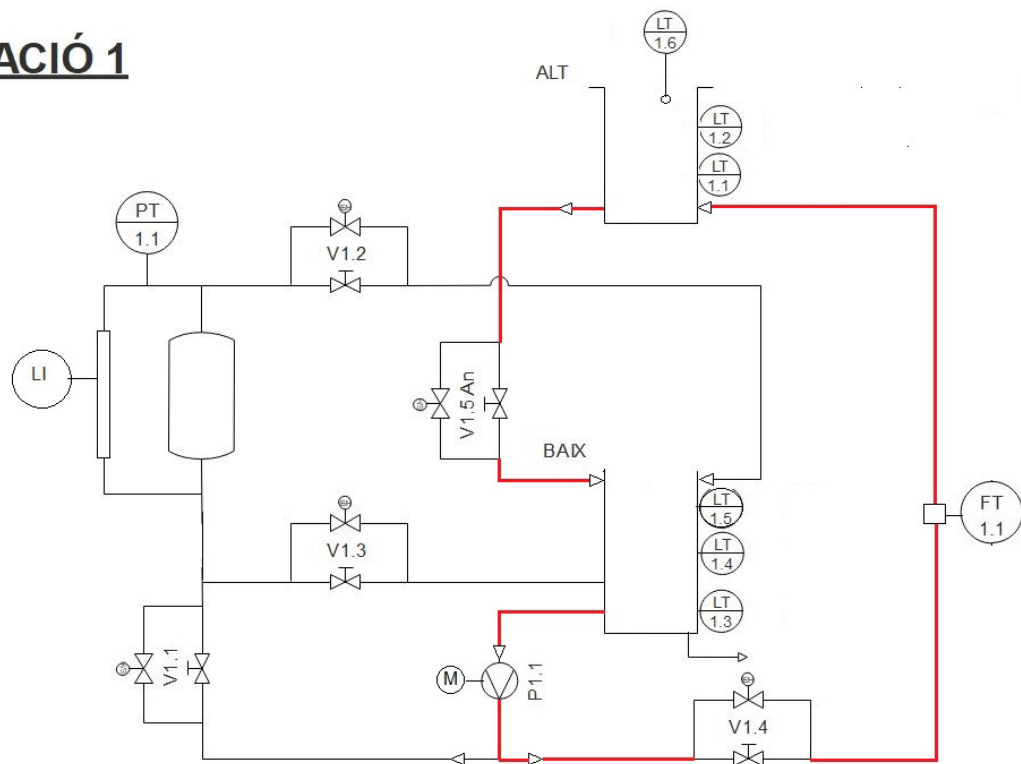
Per identificar els cables en el quadre elèctric ho fem de dues maneres segons les característiques de l'element:

- En el PLC: Aquest PLC de la sèrie X20 enumera els mòduls de sèrie d'esquerra a dreta amb una "X" i el número corresponent. Per facilitar la comprensió de les connexions, es fa servir el mateix sistema d'enumeració de mòduls acompanyat de la definició del pin al qual va connectat, per exemple si tenim un senyal d'entrada digital (DI) que la connectem en el pin 4 del mòdul X2 s'incorpora una placa d'identificació KS (amb funda d'identificació KT) amb la inscripció "X2 DI 4" tant en la connexió directa al PLC com en ambdós costats dels cables de la regleta en cas que s'hagi allargat el cable. Per saber quin cable correspon a cada transductor s'haurà de consultar les taules de connexió (apartat 4.2).
- En terminal analògic i en el variador: La identificació dels senyals en aquests elements es defineixen segons en el pin que van connectats de manera que si algun cop s'arribés a desconnectar es sàpiga on es té de reconnectar. Per conèixer els senyals del variador més fàcilment, aquests van acompanyats d'una petita definició del senyal. Per saber quin cable correspon a cada transductor caldrà consultar les taules de connexió (apartat 4.2).

3.2. Funcionalitats de la maqueta

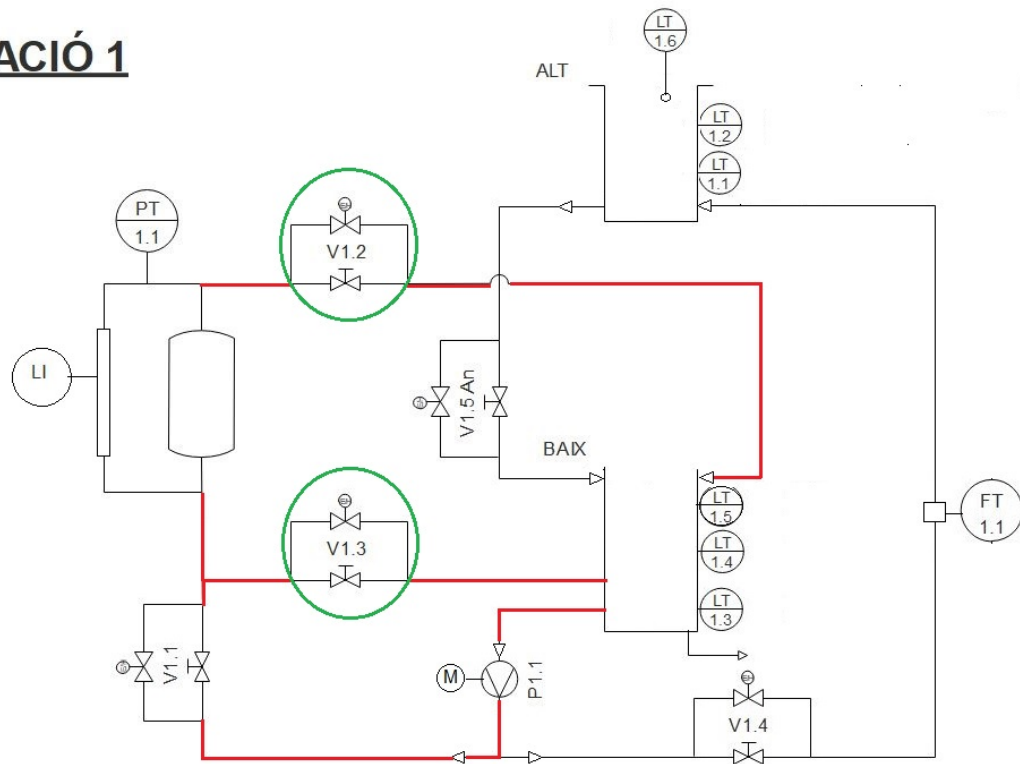
L'estació 1 incorpora el control de nivell, pressió i caudal. La maqueta té la versatilitat de poder canviar la ubicació del sensor LT1.6 podent controlar amb exactitud el nivell d'aigua que hi ha en el dipòsit inferior o superior. Hi ha la possibilitat de fer diferents controls:

- Control de nivell i cabal: Aquests controls són realitzables actuant sobre les electrovàlvules V1.4 i V1.5, i la bomba P1.1 (imatge 16), encara que el control de cabal únicament es faria regulant l'electrovàlvula V1.4 i la bomba P1.1.

ESTACIÓ 1

Imatge 16 - Circuit de control de nivell i cabal

- Control de pressió: Per tenir un control de pressió de l'estació s'han d'obrir les electrovàlvules V1.1, V1.2 i V1.3 i accionar la bomba P1. (imatge 17).

ESTACIÓ 1

Imatge 17 - Circuit de control de pressió de l'estació 1.

Els dos tipus de controls es poden ajuntar per tenir un control complet de pressió, nivell i cabal.

En el dipòsit inferior hi ha tres acoblaments per incorporar més canonades que anirien cap a futures estacions, dos d'aquests acoblaments estan dedicats a connectar-se amb l'estació distribuïdora, un serveix per agafar aigua del dipòsit BAIX i l'altre per portar aigua cap al dipòsit BAIX des d'un altra estació.

3.3. Descripció dels sensors i actuadors

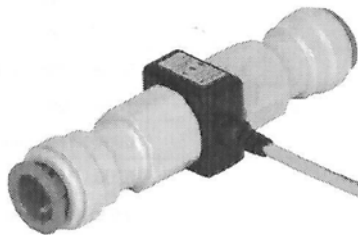
Els sensors i actuadors utilitzats dels quals es disposen són els mateixos que els de l'estació antiga.

3.3.1. Sensors

➤ Sensor de cabal FT1.1:

El sensor de cabal utilitzat és (marca), aquest sensor proporciona un senyal quadrat de freqüència de 13 a 1200 hertz i amb una tensió proporcional a la tensió d'alimentació (de 5 a 12 V). Els terminals de connexió es pot observar en l'apartat 4.2.

L'alimentació ve donada per un cable blanc (terminal positiu) i per un cable verd (terminal negatiu), arribant a consumir entre 6 i 33 mili-ampers amb un rang de temperatura de treball d'entre 0 a +65°C. Pot mesurar entre 0,5 i 15 litres / minuta amb un error d'un 1% a 20°C.

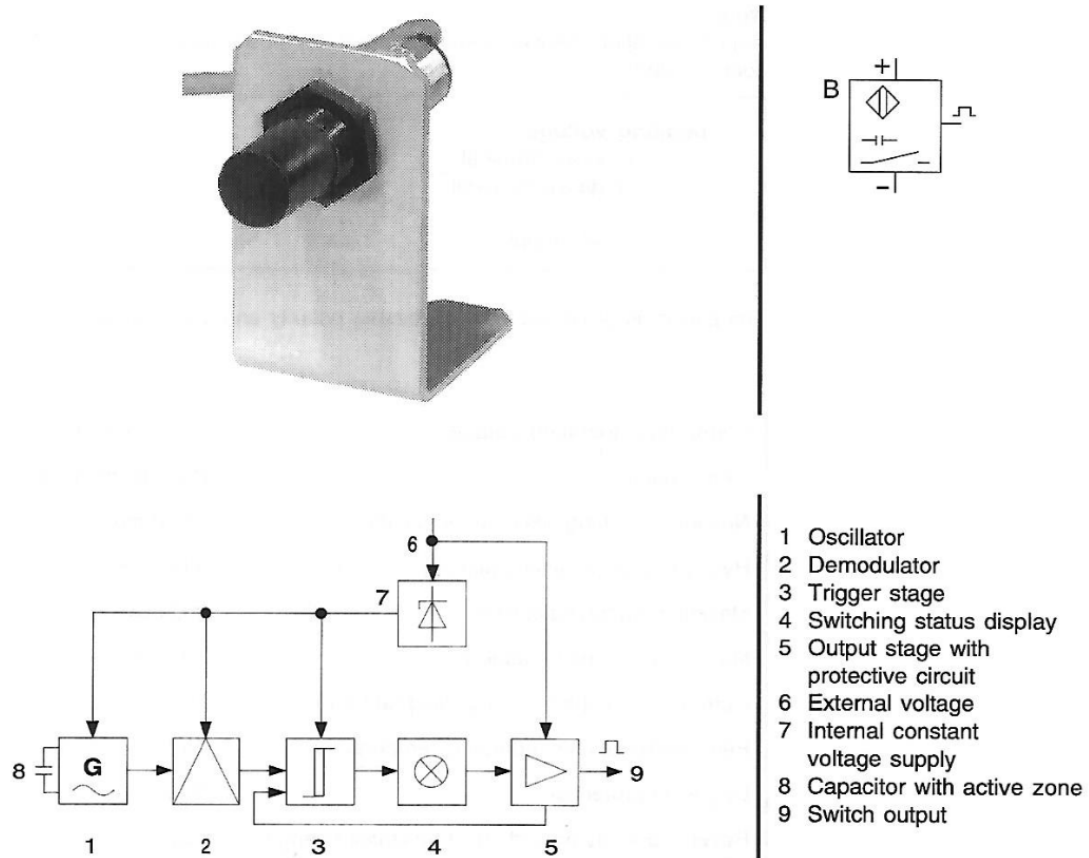


Imatge 18 - Sensor de cabal FT1.1.

➤ Sensor de nivell digital o sensor capacitiu de proximitat LT1.1 – LT1.5:

El sensor de nivell consisteix en avaluar la càrrega d'un condensador en un circuit ressonant RC, donant el senyal de sortida fent servir un contacte normalment obert (NO) amb sortida PNP. L'alimentació del sensor permet un rang de 10 a 55 volts de CC, consumint un corrent màxim de 7 mili-ampers a 55 volts amb protecció contra sobretensions. El sensor capacitiu pot arribar a fer canvis d'estat amb una freqüència de 300 Hertz a distàncies d'entre 2 a 8 mil·límetres de l'objecte, podent treballar a temperatures d'entre 20 i 70°C amb un grau de protecció IP65. El sensor té un llum verd que indica que es troba alimentat i un llum groc que indica que està donant senyal de sortida. Els terminals de connexió es pot observar en l'apartat 4.2.

El sensor capacitiu (imatge 19) té una protecció termoplàstica que incorpora una rosca de mètric 18 x 1. Està dissenyat per incorporar-se amb un angle d'alumini amb un forat passant i dues femelles, una que limita la mobilitat frontal i una altra que actua com a contrafemella.



Imatge 19 - Sensor capacitiu de proximitat LT1.1 – LT1.5 i el seu diagrama de funcionament.

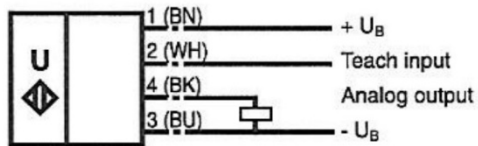
➤ **Sensor d'ultrasons o sensor de nivell analògic LT1.6 (An):**

El sensor d'ultrasons (imatge 20) treballa alimentat entre 10 i 30 volts en CC arribant a poder treballar a temperatures ambientals d'entre -24 i 70°C amb uns límits de temperatures de treball d'entre -40 i 85°C. En el cas de la sortida de senyal ve donada per un rang d'intensitat de 4 a 20 mili-ampers (cable negre) que té un rang de treball de 30 a 300 mil·límetres amb una sensibilitat de 0,4 mil·límetres i amb una freqüència de mostreig de 390 Hertz amb un retard de senyal de 30ms. El sensor incorpora un grau de protecció IP67.

El sensor permet definir el rang de distància de treball amb el senyal "teach input" (cable blanc) podent programar-lo de manera que quan el sensor detecti menys distància incrementi el senyal de sortida o a l'inrevés.

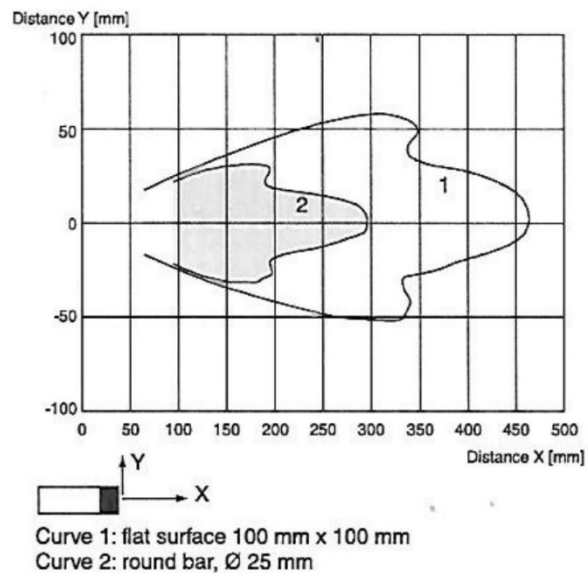


Standard symbol/Connections:
(version I)



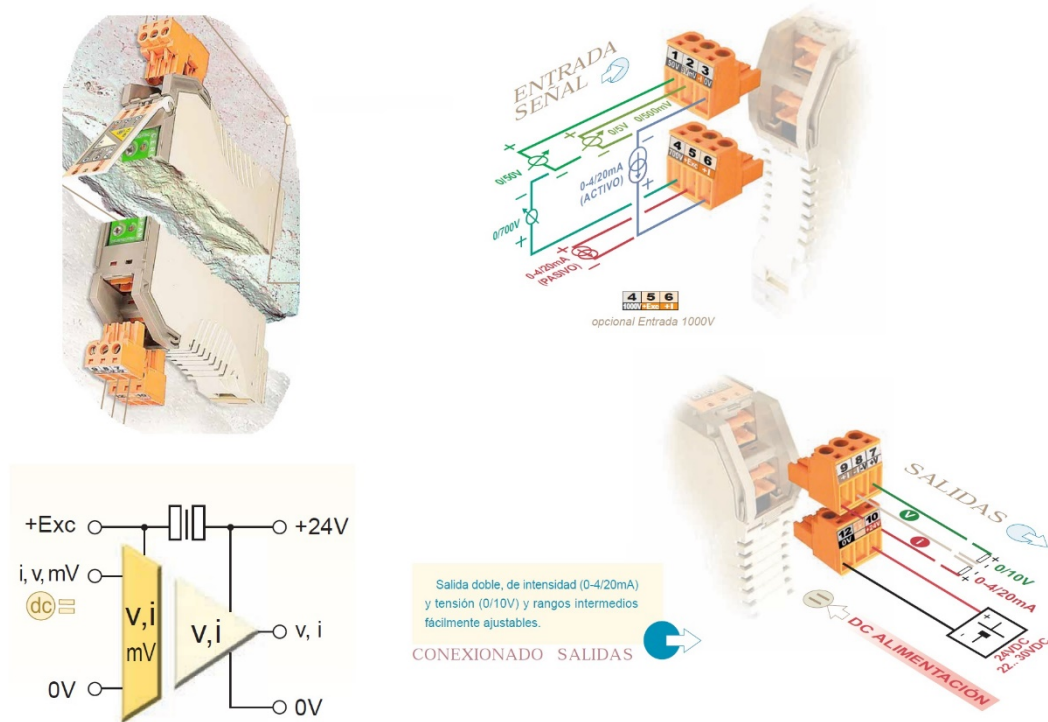
Core colors in accordance with EN 60947-5-2.

Characteristic response curve



Imatge 20 - Sensor d'ultrasons LT1.6 i la seva corba característica.

Perquè tots els senyals disposin un estàndard de control mitjançant tensió, s'incorpora un aïllador de dues vies de CC (imatge 21) que permet canviar el control d'intensitat pel control de tensió a part d'incorporar un filtre passa-baixos parametrizable per estabilitzar els senyals. En la part frontal del dispositiu hi ha el control pels senyals d'entrada, on ajustant el "CERO" es pot ajustar l'inici d'escala amb més precisió ("FINO") o amb menys precisió ("GRUESO"), així també com ajustar el final d'escala "SPAN" amb més precisió ("FINO") o amb menys precisió ("GRUESO"). Pressionant les pestanyes laterals i desplaçant frontalment es pot accedir a l'interior per seleccionar els rangs d'entrada i sortida que es volen obtenir. Els terminals de connexió es pot observar en l'apartat 4.2.

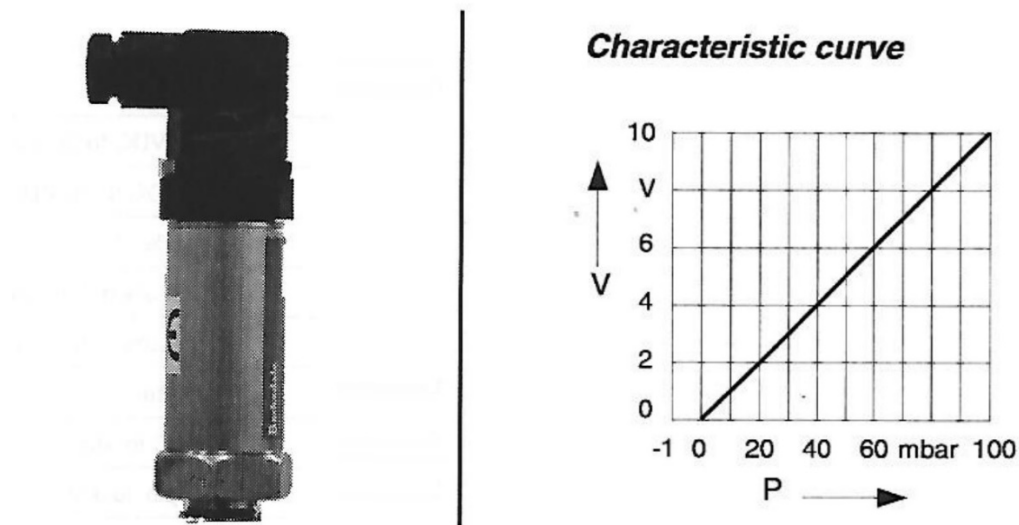


Imatge 21 - Aïllador de 2 vies, esquema de funcionament, entrades i sortides.

➤ Sensor de pressió PT1.1:

El sensor de pressió està alimentat amb un terminal positiu a 24 volts (cable marró) encara que el seu rang de tensió d'alimentació és d'entre 13 i 30 volts de CC i un terminal negatiu a 0 volts (cable blau). El senyal de sortida ve donada pel cable negre que proporciona un rang de 0 a 10 volts en un rang de pressió de 0 a 100 mili-bars, podent arribar a temperatures de treball d'entre 0 i +65°C. Els terminals de connexió es pot observar en l'apartat 4.2.

El sensor de pressió (imatge 22), fet d'alumini i acer inoxidable, incorpora un connector per a les canonades G 1/2" (equivalent a 20 mil·límetres de diàmetre). El seu funcionament es basa en un sensor piezoresistiu analògic amb un amplificador i un compensador de temperatura.



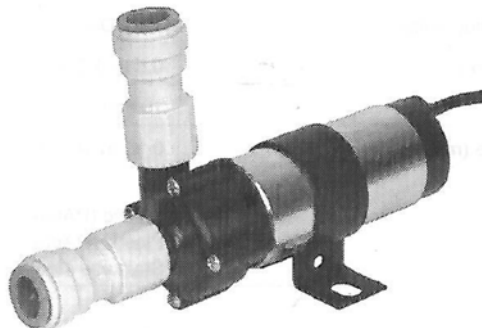
Imatge 22 - Sensor de pressió PT1.1 amb corba característica.

3.3.2. Actuadors

➤ Bomba hidràulica P1.1:

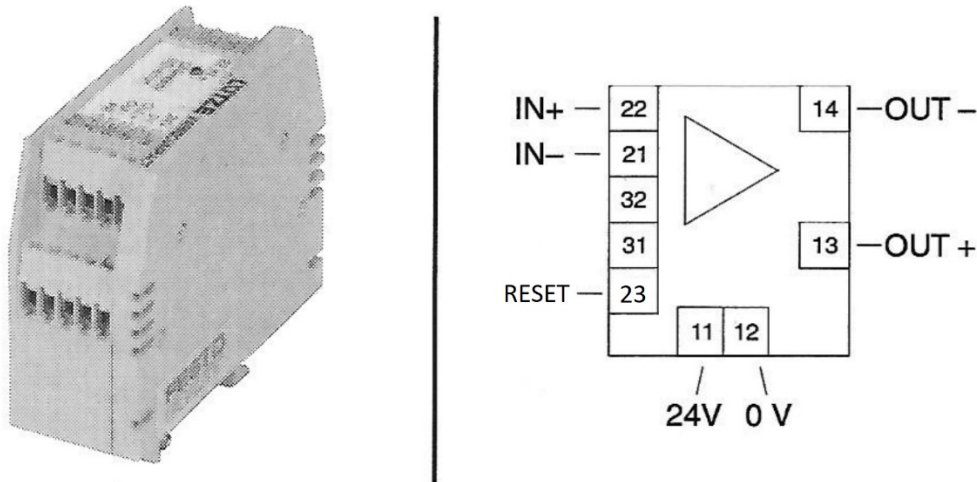
La bomba s'alimenta a 24 volts en CC, el terminal positiu ho dona el cable vermell i el negatiu ho dona pel cable negre. Consumeix un corrent de 0.5 a 0.9 ampers amb una temperatura de treball de +65°C i pot arribar a bombejar 10 litres / minut. Els terminals de connexió es pot observar en l'apartat 4.2.

La bomba hidràulica (imatge 23), feta d'acer inoxidable i plàstic, està preparada per actuar de manera contínua i pot treballar com a sistema refrigerant o amb aigua calenta. Té adaptadors amb rosca per poder adaptar connectors de canonades de 15 mil·límetres de diàmetre, on el flux d'aigua entra per la canonada horitzontal i surt per la canonada vertical.



Imatge 23 - Bomba hidràulica P1.1.

Pel control de la bomba es fa servir el variador de freqüència o regulador de motor. Permet regular la tensió d'alimentació i la velocitat de la bomba (imatge 24).

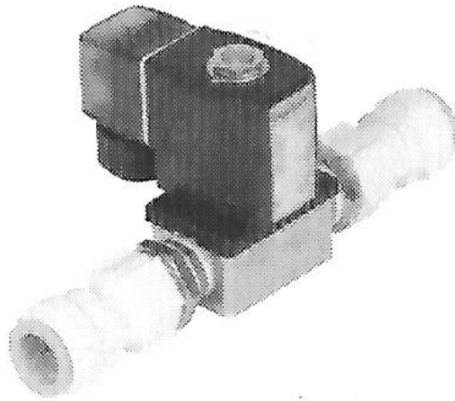


Imatge 24 - Variador de freqüència o regulador del motor i esquema I/O.

➤ **Electrovàlvules o vàlvules solenoides de dues vies V1.1 – V1.4:**

Les electrovàlvules són unes electrovàlvules que actuen directament, és a dir que quan reben tensió (24 volts en CC) la vàlvula s'obre i quan deixa de rebre tensió retorna gràcies a la força d'una molla. El cablejat de les vàlvules es basa únicament en l'alimentació, un terminal positiu i negatiu, ambdós cables negres (ja que també es pot alimentar canviant els terminals) que en aquest cas per on circulen els 24 volts s'hi ha incorporat un cable vermell i pel terminal negatiu de 0 volts un cable negre. Les electrovàlvules tenen un consum elèctric de 8 watts i arribant a temperatures de 0°C fins a +65°C podent treballar a pressions d'entre 0 i 0,5 bar. Els terminals de connexió es pot observar en l'apartat 4.2.

Les electrovàlvules (imatge 25) tenen rosca per poder adaptar connectors de canonades (connectors push-pull en aquest cas) de 15 mil·límetres de diàmetre.



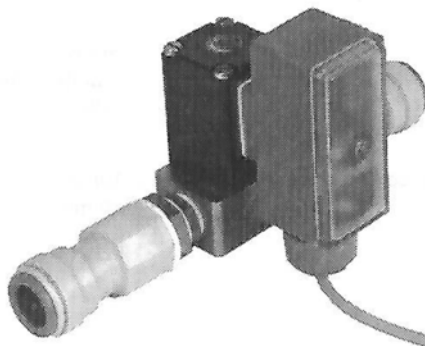
Imatge 25 - Electrovàlvula o vàlvula solenoide de dues vies V1.1 – V1.4.

➤ **Electrovàlvula analògica o proporcional V1.5 (An):**

La vàlvula proporcional pot controlar tant cabal de líquid com de gas i pot ser controlada com a element final o per un control de llaç tancat.

L'electrovàlvula s'obrirà més o menys en funció de la tensió d'entrada que se li doni en un rang de 0 a 10 volts (cable negre) i podent retornar a la posició inicial gràcies a la força d'una molla. L'electrovàlvula s'ha d'alimentar a 24 volts en corrent continu (CC) pel terminal positiu (cable marró) i a 0 volts en el pol positiu (cable blau). L'electrovàlvula proporcional consumeix 8 watts més 0.5 watts pel control electrònic que incorpora, arribant a temperatures de 0°C fins a +65°C podent treballar a pressions d'entre 0 i 0,5 bar i a una temperatura ambient de 0 a +55°C. També cal mencionar que compta amb un nivell de protecció "IP65" segons la norma de la CEI o Comissió Electrotècnica Internacional (també coneguda com IEC per les sigles en anglès). Els terminals de connexió es pot observar en l'apartat 4.2.

L'electrovàlvula analògica (imatge 26), feta de llautó, acer inoxidable i proteccions de plàstic, té rosca per poder adaptar connectors de canonades (connectors push-pull en aquest cas) de 15 mil·límetres de diàmetre.



Imatge 26 - Electrovàlvula analògica o proporcional V1.5

3.4. Teclat d'operació

Per desenvolupar el teclat d'operacions cal tenir clar els aspectes de control que es volen implementar, en aquest cas es vol un control bàsic en el teclat d'operacions físic. Per dissenyar-lo s'aprofiten alguns dels elements que incorpora l'antic teclat d'operacions. Els selectors i pulsadors tenen uns circuits encapsulats que són totalment intercanviables per triar el tipus de circuit que es vol, un circuit normalment obert (NO) o normalment tancat (NT). La tensió màxima que suporten tant els selectors com els pulsadors i les llums és de 250 volts amb un consum màxim d'1 watt.

Així doncs, els elements que formaran el teclat d'operacions són els següents (imatge 27):

- Pulsador de marxa o "START" (NO).
- Pulsador d'aturada o "STOP" (NT).
- Selector d'On / OFF .
- Selector d'Automàtic / Manual.
- Bolet d'emergència.
- Un llum blau d'alarma o "ALARM".
- Un llum verd d'encès o "POWER".
- Quatre connexions femella tipus banana amb terminal positiu (negre) i negatiu (blau) per utilitzar com a mètode d'adquisició de dades.

S'han introduït uns bornals de connexió que permeten mesurar els senyals de tensió dels elements analògics: l'electrovàlvula analògica (V1.5) i de la bomba (P1.1), i els senyals analògics del sensor de pressió (PT1.1) i del sensor d'ultrasons (LT1.6).



Imatge 27 - Teclat d'operacions.

4. SISTEMA DE CONTROL I AUTOMATITZACIÓ

El sistema de control i automatització de la maqueta es basa en 2 PLCs de diferents característiques de l'empresa *B&R Industrial Automation GmbH*. Els PLC's d'aquesta empresa fan servir l'entorn de programació "Automation Studio", una eina ideal per a la programació amb diferents tipus de llenguatge incloent els llenguatges de programació de la norma IEC 61131-3.

4.1. PLC's de la maqueta

L'empresa *B&R Industrial Automation GmbH* ha equipat als seus dispositius de la última tecnologia per adaptar-se al marc de l'indústria 4.0, incorporant tecnologies com POWER-LINK (per la comunicació a temps real), capacitat multitasca i programació amb Automation Studio de B&R (on es poden utilitzar tots els llenguatges IEC6113-3 i C).

Els PLCs que s'han utilitzat a la maqueta fan servir el sistema "X20", un sistema nascut de la filosofia de l'empresa "Perfection in automation" i adquirida a base de l'experiència en aplicacions, que permet controlar des d'aplicacions simples a les aplicacions més exigents arribant a gestionar la informació en l'ordre de 200 microsegons.

Els seus PLCs incorporen un sistema d'entrades i sortides modular, arribant a connectar directament fins a 250 mòduls de E/S X20. Cada mòdul està format per 3 subcomponents: bloc terminal, mòdul electrònic i mòdul bus. Cada mòdul té una amplada de 12,5mm on hi poden arribar a haver-hi 12 canals en un sol mòdul, optimitzant el sistema de E/S de manera que té un 50% més de capacitat de E/S que un PLC convencional.

Els PLCs que estan muntats en les estacions són el X20 CP 1583 en l'estació distribuïdora i el X20 CP 1381 en les estacions de control de nivell, pressió i cabal

4.1.1. X20 CP 1583

Aquest PLC prové de la família X20(c)CP158x i X20(c)CP358x. Aquests PLCs estan basats en la tecnologia de processador Intel® ATOM™ de vanguardia, les CPUs X20 cobreixen un ampli espectre de requisits. Es poden implementar en solucions que van des d'aplicacions estàndard fins a aquells que requereixen els nivells més alts de rendiment.

La interfície Ethernet estàndard és capaç de manejar la comunicació en el rang de gigabits. Per a un rendiment encara més gran de la xarxa en temps real, la interfície POWERLINK és compatible amb el mètode d'encadenament de resposta d'enquestes (PRC).

Aquest PLC modular té els següents aspectes tècnics:

- CPU X20, compatibilitat amb Atom 333 MHz.
- Memòria RAM DDR2 de 128 MB, SRAM d'1 MB
- Memòria desmuntable: CompactFlash.
- 1 ranura d'inserció per a mòduls d'interfície X20.
- 2 interfícies USB (IF4 i IF5), 1 interfície RS232, 1 interfície Ethernet 10/100 / 1000 BASE-T, 1 interfície POWERLINK.
- Incorpora un mòdul d'alimentació.
- 1 bloc de terminals X20TB12 amb coberta X20AC0SR1.
- Consum de 4.8 Watts sense mòduls d'interfície i USB.

El PLC incorpora un petit panell amb llums LED que indica l'estat de la CPU (imatge 28) on els diferents estats estan especificats en la taula 2.



Imatge 28 - Panell d'estat de la CPU del PLC X20CP1583.

Taula 2 - Taula d'estats de la CPU del PLC X20CP1583.

LED	Color	Status	Description
R/E	Green	On	Application running
		Blinking	Boot mode system start: CPU initializing the application, all bus systems and I/O modules ¹⁾
		Double flash	BOOT mode (during firmware update) ¹⁾
	Red	On	SERVICE mode
		Blinking	The "R/E" LED blinks red and the "RDY/F" LED blinks yellow when there is a license violation.
RDY/F	Yellow	On	SERVICE or BOOT mode
		Blinking	The "RDY/F" LED blinks yellow and the "R/E" LED blinks red when there is a license violation.
S/E	Green/Red		Status/Error LED. The statuses of this LED are described in section "LED "S/E"" on page 10.
PLK	Green	On	A link to the POWERLINK peer station has been established.
		Blinking	A link to the POWERLINK peer station has been established. The LED blinks when Ethernet activity is taking place on the bus.
ETH	Green	On	A link to the peer station has been established.
		Blinking	A link to the peer station has been established. Indicates Ethernet activity is taking place on the bus.
CF	Green	On	CompactFlash inserted and detected
	Yellow	On	CompactFlash read/write access
DC	Yellow	On	CPU power supply OK
	Red	On	Backup battery empty

El sistema de comunicacions "POWERLINK" també té dos llums LED indicant els diferents estats en que es pot trobar.

Taula 3 - Taula d'estats del sistema de comunicacions "POWERLINK".

Status LED		Status of the POWERLINK node
Green	Red	
On	Off	The POWERLINK node is running with no errors.
Off	On	A system error has occurred. The type of error can be read using the PLC logbook. An irreparable problem has occurred. The system cannot properly carry out its tasks. This state can only be changed by resetting the module.
Blinking alternately		The POWERLINK managing node has failed. This error code can only occur when operated as a controlled node. This means that the configured node number lies within the range 0x01 - 0xFD.
Off	Blinking	System stop. The red blinking LED signals an error code (see "System failure error codes" on page 12).
Off	Off	Module is: <ul style="list-style-type: none"> Switched off Starting up Not configured correctly in Automation Studio Defective

Per conèixer l'estat de l'alimentació incorpora un segon panell que té uns indicadors LED tal com mostra la imatge 29 i la taula 4.

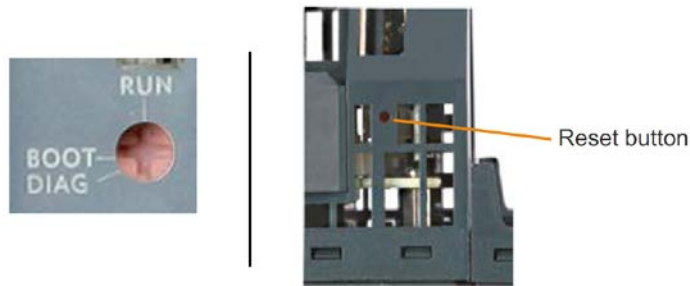


Imatge 29 - Panell d'estat de l'alimentació del PLC X20CP1583.

Taula 4 - Taula d'estats del sistema d'alimentació del PLC X20CP1583.

LED	Color	Status	Description
r	Green	Off	No power to module
		Single flash	RESET mode
		Blinking	PREOPERATIONAL mode
		On	RUN mode
e	Red	Off	No power to module or everything OK
		Double flash	LED indicates one of the following states: <ul style="list-style-type: none">• The X2X Link supply for the power supply is overloaded• I/O supply too low• Input voltage for X2X Link supply too low
e + r	Red on / Green single flash		Invalid firmware
S	Yellow	Off	No RS232 activity
		On	The LED lights up when data is being sent or received via the RS232 interface.
l	Red	Off	The X2X Link supply is within the valid limits
		On	The X2X Link supply for the power supply is overloaded

El PLC incorpora un selector de mode d'operació que es mostra en la imatge 30 amb diversos modes indicats en la taula 5. També incorpora un botó de “Reset” que permet aturar tots els programes així com anul·lar el valor de totes les sortides (imatge 30).

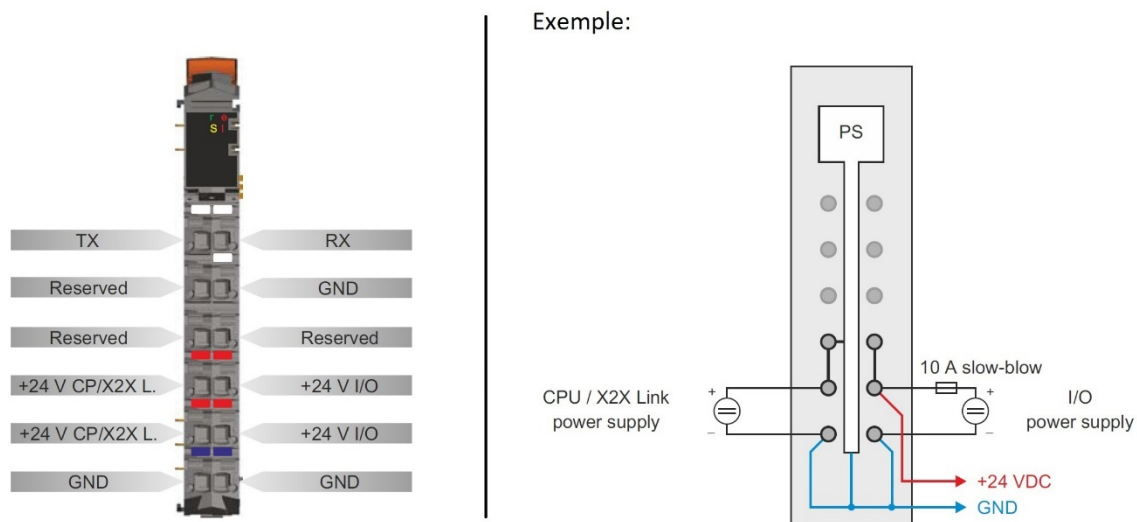


Imatge 30 - Selector de mode i botó de "Reset" del PLC X20CP1583.

Taula 5 - Taula de modes d'operació del PLC X20CP1583.

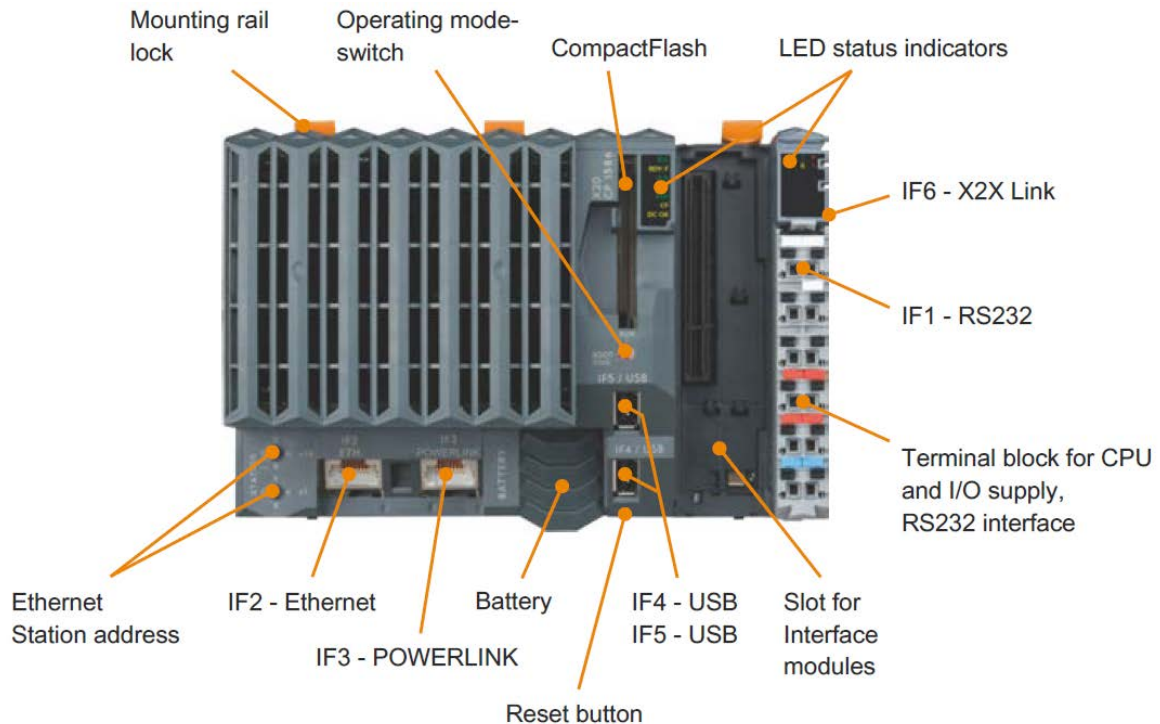
Switch position	Description
BOOT	When the switch is in this position, the default B&R Automation Runtime (AR) system is started and the runtime system can be installed via the online interface (B&R Automation Studio). User flash memory is deleted only after the download begins.
RUN	RUN mode
DIAG	Boots the CPU in diagnostic mode. Program sections in User RAM and User FlashPROM are not initialized. After diagnostic mode, the CPU always boots with a cold restart.

Una font d'alimentació està integrada a la CPUs X20. S'alimenta directament per la CPU, X2X Link i la font d'alimentació interna d'E / S. Les línies d'alimentació de la CPU / X2X Link estan aïllades elèctricament. El mateix mòdul incorpora en els terminals 1, 2 i 4 una interfície de comunicació RS232 aïllada elèctricament perquè s'utilitzi com a interfícies de comunicació online amb el PLC programat (imatge 31).



Imatge 31 - Mòdul d'alimentació i de comunicació RS232 del PLC X20CP1583.

En la imatge 32 es pot veure com estan distribuïts els diferents elements del mòdul X20CP1583.



Imatge 32 - PLC X20CP1583.

Per obtenir més informació sobre el PLC X20CP1583 accedir a la web oficial de B&R¹.

4.1.2. X20 CP 1381

El PLC X20CP1381 prové de la família X20(c)CP1301, X20CP1381 i X20CP1382. En aquests PLC, tant per USB com per Ethernet com les interfícies d'autobús integrades POWERLINK i CAN, permeten diversificar en el tipus de connexions que es puguin necessitar. Si es necessiten connexions addicionals de bus de camp, la CPU es pot actualitzar amb un mòdul d'interfície des del rang estàndard de productes X20. Aquestes CPU no requereixen ventiladors o bateries i, per tant, són lliures de manteniment. S'inclouen 30 entrades i sortides digitals diferents i 2 entrades analògiques en els dispositius. Es pot utilitzar una entrada analògica per a la mesura de la temperatura de resistència PT1000. Les característiques del PLC X20CP1381 són les següents:

- CPU X20 és un Intel x86 200Hz amb un processador integrat per les I / O.
- Memòria SRAM 128 MB DDR3.

¹ <https://www.br-automation.com/es/productos/sistemas-de-control/sistema-x20/cpus-x20/x20cp1583/>

- Memòria flash d'entrada a 1 GB
- unitat, ranura d'inserció per a mòduls d'interfície X20,
- 2 interfícies USB, 1 interfície RS232, 1 interfície d'autobús CAN, 1 interfície POWER-LINK i 1 interfície Ethernet 10 / 100BASE-T.
- 14 entrades digitals (24V i 0,5A).
- 2 entrades analògiques per senyals de $\pm 10V$ o de 0 a 20mA o de 4 a 20mA.
- 4 sortides digitals (24V i 0,2A).
- 4 entrades / sortides digitals programables (24V i 0,5A).
- Inclou un mòdul d'alimentació.
- 1 PT1000 en comptes d'una entrada analògica.
- 3 terminals X20TB1F amb coberta X20AC0SR1.
- Consum de 4,8 Watts sense mòduls d'interfície i USB.

➤ Mòdul X1

El PLC incorpora un petit panell amb llums LED que indica els estats del mòdul en concret. El primer panell dona la informació sobre el primer mòdul X1, on indica l'estat de la CPU a més de l'estat de les entrades digitals del mòdul X1 (imatge 33). Per conèixer els estats del mòdul X1 mirar la taula 6.



Imatge 33 – Panell del mòdul X1 del PLC X20CP1381.

Taula 6 - Taula d'estats de la CPU del PLC X20CP1581.

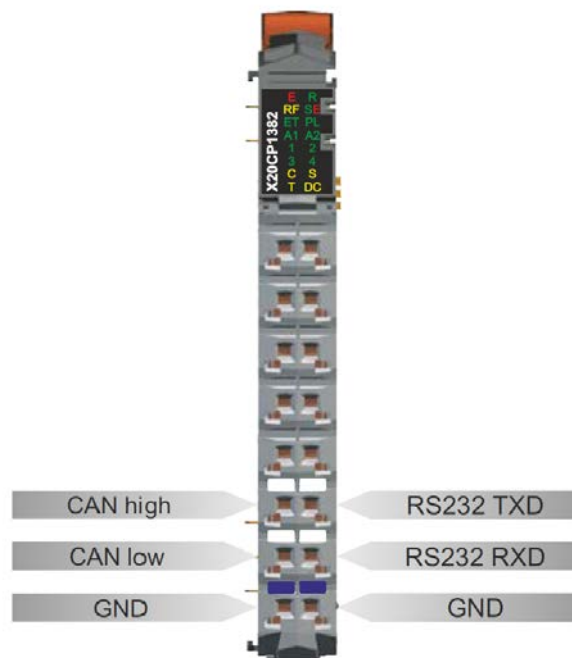
LED	Color	Status	Description
E	Red	On	SERVICE or BOOT operating mode
		Blinking	The "E" LED blinks red and the "RF" LED blinks yellow when there is a license violation.
		Double flash	Firmware update ¹⁾
R	Green	On	Application running
		Blinking	Boot mode system start: CPU initializing the application, all bus systems and I/O modules ¹⁾
	Red	On	During reset
RF	Yellow	On	SERVICE or BOOT operating mode
		Blinking	The "RF" LED blinks yellow and the "E" LED blinks red when there is a license violation.
SE	Green/Red		Status/Error LED. The statuses of this LED are described in section ""S/E" LED" on page 9.
ET	Green	On	A link to the Ethernet remote station has been established.
		Blinking	A link to the Ethernet remote station has been established. The LED blinks when Ethernet activity is taking place on the bus.
PL	Green	On	A link to the POWERLINK peer station has been established.
		Blinking	A link to the POWERLINK peer station has been established. The LED blinks when Ethernet activity is taking place on the bus.
A1 - A2	Green	Off	Open circuit or disconnected sensor
		Blinking	Input signal overflow or underflow
		On	Analog/digital converter running, value OK
1 - 4	Green		Input state of the corresponding digital input
C	Yellow	On	CPU transmitting or receiving data via the CAN bus interface
S	Yellow	On	CPU transmitting or receiving data via the RS232 interface
T	Yellow	On	The terminating resistor integrated in the CPU is switched on.
DC	Yellow	On	CPU power supply OK

El sistema de comunicacions "POWERLINK" també té dos llums LED indicant els diferents estats en els quals es pot trobar, depenent de la freqüència del parpelleig ens indicarà un estat o un altre (mirar taula 7).

Taula 7 - Taula d'estats de la comunicació POWERLINK del PLC X20CP1581.

Green - Status	Description
Off	<p>Mode The module is in mode NOT_ACTIVE or:</p> <ul style="list-style-type: none"> Switched off Starting up Not configured correctly in Automation Studio Defective <p>Managing node (MN) The bus is being monitored for POWERLINK frames. If a corresponding frame is not received within the defined time frame (timeout), then the module switches immediately to mode PRE_OPERATIONAL_1. If POWERLINK communication is detected before the time expires, however, then the MN will not be started.</p> <p>Controlled node (CN) The bus is being monitored for POWERLINK frames. If a corresponding frame is not received within the defined time frame (timeout), then the module switches immediately to mode BASIC_ETHERNET. If POWERLINK communication is detected before this time expires, however, the module switches immediately to mode PRE_OPERATIONAL_1.</p>
Flickering green (approx. 10 Hz)	<p>Mode The module is in mode BASIC_ETHERNET. The interface is being operated as an Ethernet TCP/IP interface.</p> <p>Managing node (MN) This state can only be changed by resetting the module.</p> <p>Controlled node (CN) If POWERLINK communication is detected while in this state, the module will transition to state PRE_OPERATIONAL_1.</p>
Single flash (approx. 1 Hz)	<p>Mode The module is in mode PRE_OPERATIONAL_1.</p> <p>Managing node (MN) The MN starts "reduced cycle" operation. Cyclic communication is not yet taking place.</p> <p>Controlled node (CN) The module can be configured by the MN in this state. The CN waits until it receives an SoC frame and then switches to mode PRE_OPERATIONAL_2. An LED lit red in this state indicates failure of the MN.</p>
Double flash (approx. 1 Hz)	<p>Mode The module is in mode PRE_OPERATIONAL_2.</p> <p>Managing node (MN) The MN begins cyclic communication (cyclic input data is not yet being evaluated). The CNs are configured in this state.</p> <p>Controlled node (CN) The module can be configured by the MN in this state. A command then switches the module to mode READY_TO_OPERATE. An LED lit red in this mode indicates failure of the MN.</p>
Triple flash (approx. 1 Hz)	<p>Mode The module is in state READY_TO_OPERATE.</p> <p>Managing node (MN) Cyclic and asynchronous communication is taking place. Any received PDO data is ignored.</p> <p>Controlled node (CN) The configuration of the module is completed. Normal cyclic and asynchronous communication is taking place. The transmitted PDO data corresponds to the PDO mapping. Cyclic data is not yet being evaluated, however. An LED lit red in this mode indicates failure of the MN.</p>
On	<p>Mode The module is in mode OPERATIONAL. PDO mapping is active and cyclic data is being evaluated.</p>
Blinking (approx. 2.5 Hz)	<p>Mode The module is in mode STOPPED.</p> <p>Managing node (MN) This status is not possible for the MN.</p> <p>Controlled node (CN) No output data is produced or input data supplied. It is only possible to enter or leave this mode after the MN has given the appropriate command.</p>

També incorpora una interfície RS232 no aïllada elèctricament que té com a principal objectiu servir com a interfície en línia per comunicar-se amb el dispositiu de programació. A excepció de la CPU del nivell d'entrada, totes les CPU compactes estan equipades amb una interfície d'autobús CAN no aïllada elèctricament. Aquests sistemes de comunicació estan ubicats en la part inferior de les I/O del mòdul X1 (imatge 34).



Imatge 34 - Entrades i sortides de les interfícies de comunicació CAN i RS232 del PLC X20CP1581 (mòdul X1).

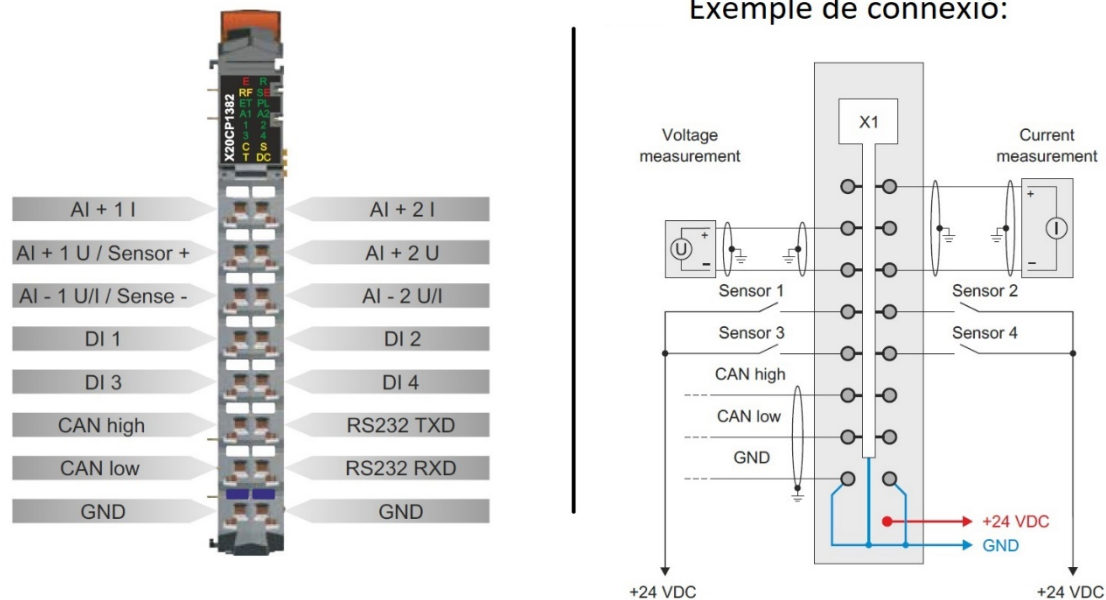
Una resistència està instal·lada en el mòdul X1 per a la resistència de finalització de bus CAN (imatge 35). Es pot encendre i apagar amb un interruptor a la part superior del mòdul X1. La "T" LED indica una resistència de tancament activa.



Imatge 35 - Interruptor per habilitar / deshabilitar la resistència de finalització de bus CAN del PLC X20CP1581 (mòdul X1).

El mòdul X1 incorpora 4 entrades analògiques: dues entrades analògiques de corrent (AI+1 I, AI+2 I) en els dos primers pins (en horitzontal), la segona filera de pins incorpora dues entrades analògiques de tensió (AI+1U, AI+2U) on en la tercera fila de pins s'incorpora els

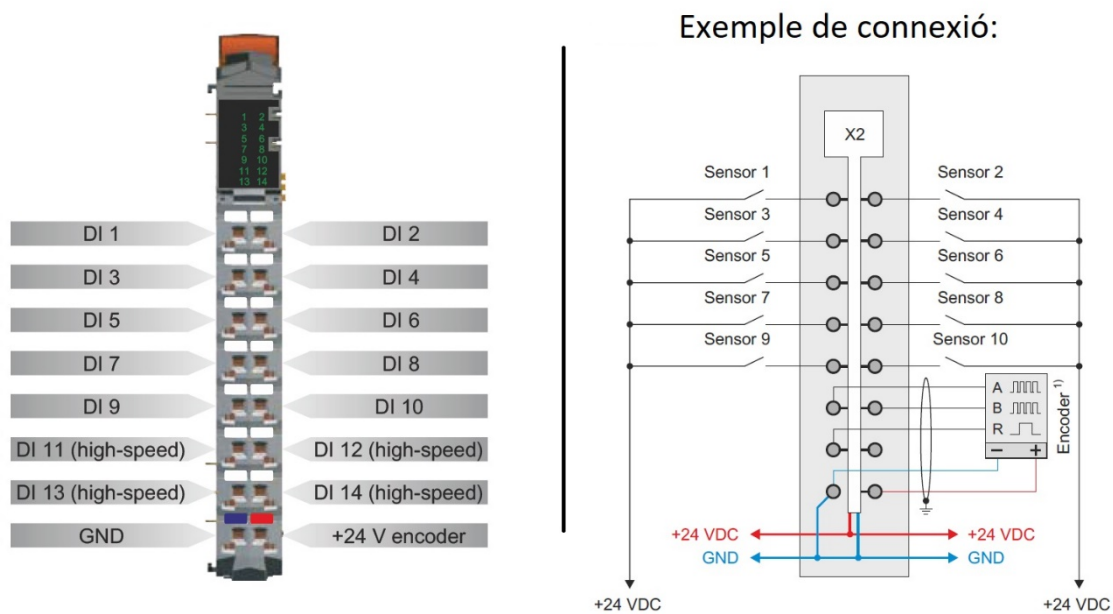
terminals negatius de les entrades analògiques. La quarta i cinquena fila de pins té 4 entrades digitals (DI1 a DI4) (imatge 36).



Imatge 36 - Pinout i exemple del mòdul X1 del PLC X20CP1581.

➤ Mòdul X2

El mòdul X2 incorpora 14 entrades digitals: 10 entrades digitals estan ubicades en les primeres cinc files de pins, les altres 4 entrades digitals d'alta velocitat. L'última fila de pins serveix per alimentar el mòdul que en cas de que un altre mòdul ja estigui alimentat, aquesta alimentació es comuna per als demás mòduls (imatge 37).



Imatge 37 - Pinout i exemple del mòdul X2 del PLC X20CP1581.

És important tenir en compte que només es poden fer dues funcions del mateix tipus en l'eix vertical (de la DI11 a la DI14, penúltima i antepenúltima fila de pins).

Taula 8 - Possibles funcions de les entrades digitals de comptatge ràpid DI11 a DI14 del mòdul X2 del PLC X20CP1581.

Channel	Counter function	Edge detection ¹⁾				
DI 11	Event counter 1	A	A	D - Direction	<ul style="list-style-type: none">Period measurementGate measurementDifferential time measurement	<ul style="list-style-type: none">Edge countersEdge times
DI 12		B	B	F - Frequency	<ul style="list-style-type: none">Period measurementGate measurementDifferential time measurement	<ul style="list-style-type: none">Edge countersEdge times
DI 13	Event counter 2	A	R	R	<ul style="list-style-type: none">Period measurementGate measurementDifferential time measurement	<ul style="list-style-type: none">Edge countersEdge times
DI 14		B	E - Reference enable	E - Reference enable	<ul style="list-style-type: none">Period measurementGate measurementDifferential time measurement	<ul style="list-style-type: none">Edge countersEdge times

Cal tenir en compte els següents punts per configurar correctament les entrades digitals d'alta velocitat:

- Les funcions de comptador són mútuament excloents. Només es pot seleccionar un tipus de funció de comptador a la vegada. No és possible seleccionar dos comptadors d'esdeveniments (DI 11 i DI 13) al mateix temps juntament amb un comptador AB o DF (cada un en DI 13 i DI 14)!
- És possible seleccionar una funció de comptatge i detecció de flanc al mateix temps.
- Una posició o comptador de flanc és possible quan es configuren les entrades d'alta velocitat com a comptador d'esdeveniments x2, codificador incremental ABR o funció DF.

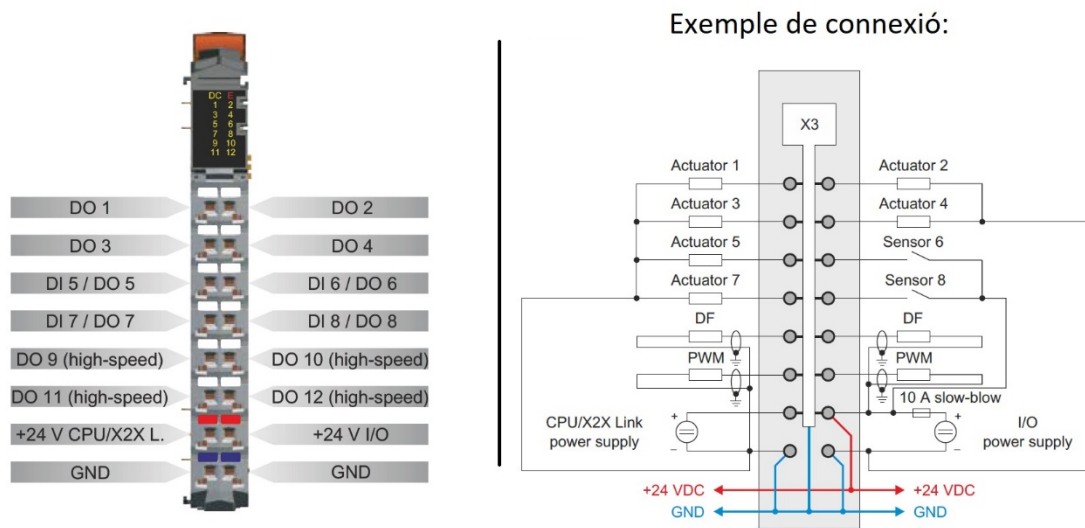
Per saber si les entrades digitals donen senyal al mòdul, incorpora un petit panell on indica l'estat de cada entrada digital amb un llum verd si està actiu (imatge 38).



Imatge 38 - Panell del mòdul X2 del PLC X20CP1381.

➤ Mòdul X3

El mòdul X3 incorpora 4 sortides digitals (DO1 a DO4 en les dues primeres files de pins), 4 entrades i sortides digitals programables (DI5/DO5 a DI8/DO8 en la tercera i quarta files de pins), 4 sortides digitals d'alta velocitat (DO9 a DO12), la penúltima fila de pins hi trobem una alimentació comuna de la CPU i de les entrades i sortides (+24V I/O) i l'última fila el terminal negatiu comú (GND) (imatge 39). Per evitar la interferència, cada línia de senyal de les sortides digitals d'alta velocitat ha d'estar protegida individualment. La longitud màxima permesa del cable és de 20 m.



Imatge 39 - Pinout i exemple del mòdul X3 del PLC X20CP1581.

Cal tenir en compte que els canals 5 a 8 estan dissenyats com a canals mixtos. Si s'utilitza un d'aquests canals, és absolutament necessari assegurar-se que no hi hagi una tensió externa en el canal I/O quan es talli la font d'alimentació de I/O. En cas contrari, la potència retornarà al terminal positiu de la font d'alimentació de I/O. Això provocarà components defectuosos.

Les següents solucions estan disponibles per evitar que es produeixi el retorn de la potència:

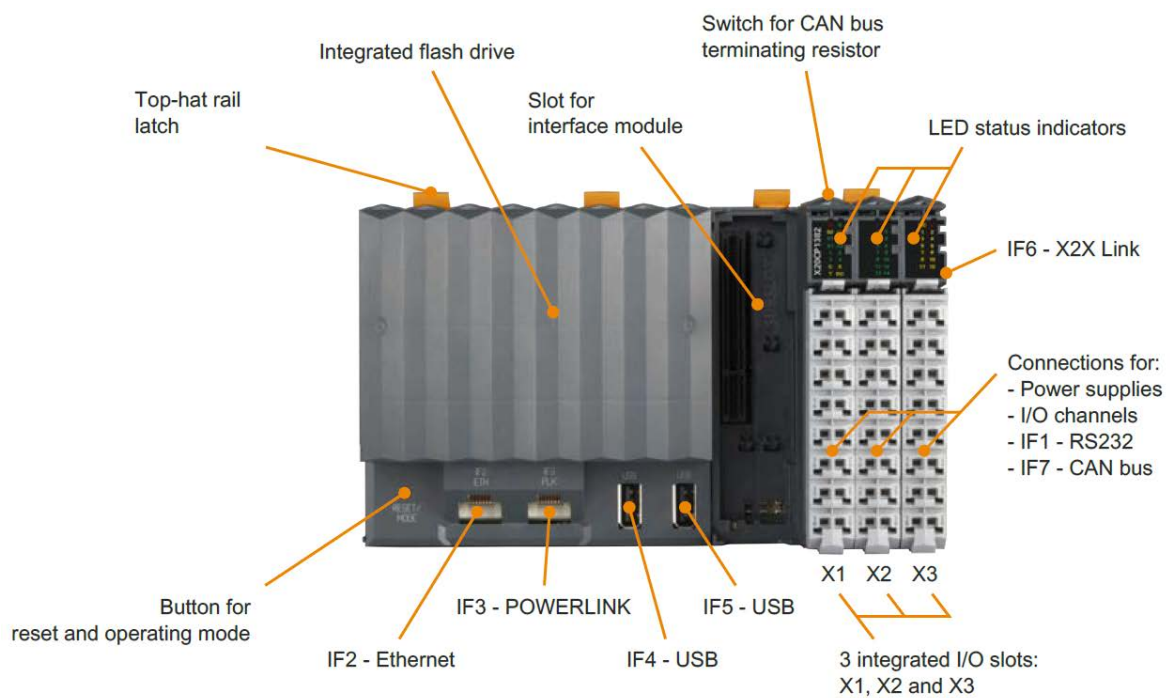
- No es pot apagar la font d'alimentació de I/O de la CPU, ja que permet mantenir el potencial de referència.
- Si la font d'alimentació de I/O s'activa de totes maneres (p. Ex., Com a part de la cadena E-stop), també s'ha de desactivar el subministrament dels sensors i/o actuadors. Això evita el possible retorn de potència i protegeix els components de la seva destrucció.

Per saber l'estat de les I/O del mòdul així també com les sortides d'alta velocitat, incorpora un petit panell on indica els estats de cada entrada / sortida amb un llum groc. També incorpora un símbol "DC" per saber l'estat de la tensió d'alimentació i un símbol "E" que ens indicarà fent pampallugues quan no hi ha alimentació del mòdul (imatge 40).



Imatge 40 - Panell del mòdul X3 del PLC X20CP1381.

En la imatge 41 es pot veure com estan distribuïts els diferents elements del PLC X20CP1381.



Imatge 41 - PLC X20CP1381.

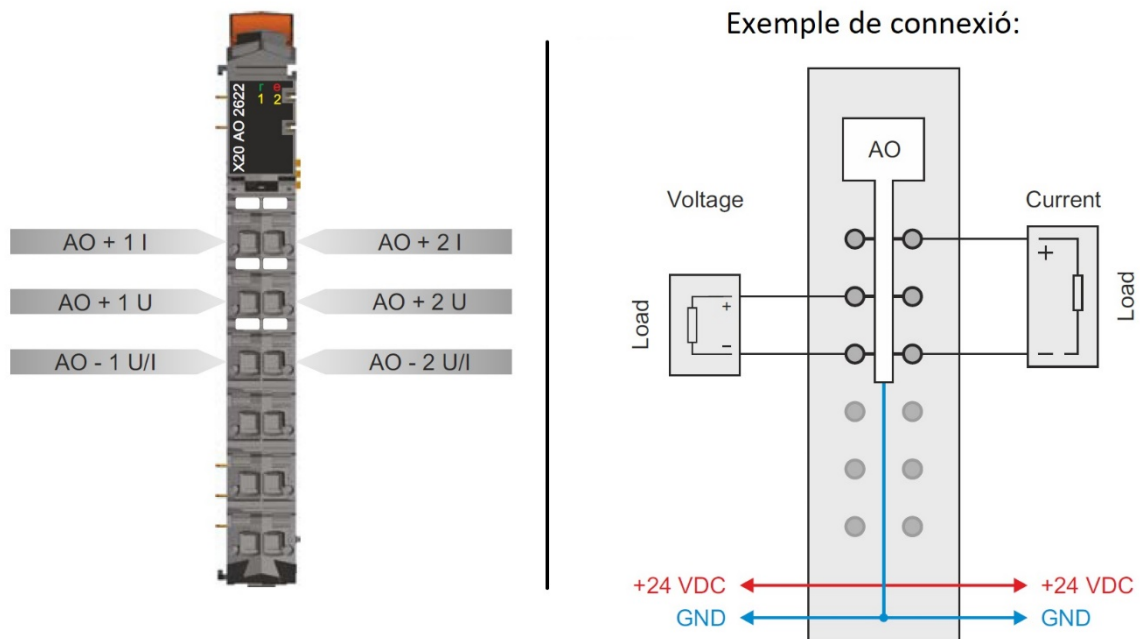
Per obtenir més informació sobre el PLC X20CP1381 accedir a la pagina oficial de B&R².

²<https://www.br-automation.com/es/productos/sistemas-de-control/sistema-x20/cpus-x20/x20cp1381/#basicinfo>

➤ Mòdul agregat X4

El mòdul X4 està equipat amb 2 sortides amb resolució de convertidor digital de 13 bits (inclòs el signe) (imatge 42). És possible seleccionar entre senyal de corrent i de tensió fent servir diferents terminals. Aquest mòdul està dissenyat per a blocs de terminals X20 de 6 pins. Si és necessari, també es pot utilitzar el bloc de terminals de 12 pins.

- 2 sortides analògiques de corrent o tensió.
- Consum de 1,1 Watts de gestió d'entrades i sortides.



Imatge 42 - Pinout i exemple del mòdul X4 del PLC X20CP1581.

Per saber l'estat del mòdul hi ha un panell frontal amb llums LED que ens indica l'estat actual de cada pin i l'estat general del mòdul (imatge 43). Aquests estats es poden observar en la taula .



Imatge 43 - Panell del mòdul X4 del PLC X20CP1381.

Taula 9 - Estats del mòdul X4 del PLC X20CP1381.

Color	Status	Description
Green	Off	No power to module
	Single flash	RESET mode
	Double flash	BOOT mode (during firmware update) ¹⁾
	Blinking	PREOPERATIONAL mode
	On	RUN mode
Red	Off	No power to module or everything OK
	On	Error or reset status
Red on / Green single flash		Invalid firmware
Orange	Off	Value = 0
	On	Value ≠ 0

Per obtenir més informació sobre el mòdul X20AO2632 accedir a la pagina oficial de B&R³.

4.2. Connexions amb els sensors i actuadors

Les connexions amb els diferents sensors i actuadors es troben en les següents taules.

Taula 10 – Taula de connexions aïllador de 2 vies.

AÏLLADOR DE 2 VIES			
PIN	Sigles	CONNEXIÓ	CABLE
3	I -	(-) LT1.6	blau
6	I +	(senyal +) LT1.6	negre
7	V +	AI+2U	gris
8	V -	AI-2U/I	negre
10	(+24V)	Alimentació I/O	vermell
12	(0V)	GND I/O	blau

³ <https://www.br-automation.com/es/productos/sistemas-de-control/sistema-x20/modulos-de-salidas-analogicas/x20ao2632/#basicinfo>

Taula 11 - Taula de connexions Variador.

VARIADOR			
PIN	Sigles	CONNEXIÓ	CABLE
11	(0V)	GND I/O	negre
12	(+24V)	Alimentació I/O	negre
13	OUT -	(-) P1.1	negre
14	OUT+	(senyal+) P1.1	vermell
21	IN -	AO-2U/I	blau
22	IN +	AO+2U	blau
23	RESET	PIN 23 TERMINAL AN.	Blau

Taula 12 - Taula de connexions terminal analògic.

TERMINAL ANALÒGIC		
PIN	CONNEXIÓ	CABLE
04 (+10V)	(+) LT1.6	marró
06 (0V)	(-) LT1.6	blau
7 (+10V)	(+) FT1.1	groc
9 (0V)	(-) FT1.1	verd
16 (+24V)	(+) PT1.1	marró
18 (0V)	(-) PT1.1	blau
19 (+24V)	Alimentació I/O	vermell
20 (+24V)	(+) LT1.1	marró
21 (+24V)	(+) LT1.2	marró
22 (+24V)	(+) LT1.3	marró
regleta(+24V)	(+) LT1.4	negre
regleta (+24V)	(+) LT1.5	negre
regleta (GND)	(-) LT1.4	blau
regleta (GND)	(-) LT1.5	blau
23 (0V)	PIN 23 VARIADOR (RESET)	blau
25 (0V)	PIN 12 GND VARIADOR	blau
28 (0V)	GND TERMINAL AN.	blau
29 (0V)	(-) V1.1	negre
30 (0V)	(-) V1.2	negre
31 (0V)	(-) V1.3	negre
32 (0V)	(-) V1.4	negre
34(0V)	(-) LT1.1	blau
36(0V)	(-) LT1.2	blau
38(0V)	(-) LT1.3	blau

Taula 13 - Taula de connexions Mòdul X1.

MÒDUL X1 CPU X20 CP 1381			
PIN	Sigles	CONNEXIÓ	CABLE
2	AI+1U	PIN 14 VARIADOR	gris
3	AI+2U	PIN 7 AÏLLADOR	gris
4	AI-1U/I	PIN 13 VARIADOR	gris
5	AI-2U/I	PIN 8 AÏLLADOR	negre

Taula 14 - Taula de connexions mòdul X2.

MÒDUL X2 CPU X20 CP 1381			
PIN	Sigles	CONNEXIÓ	CABLE
1	DI1	LT1.1	negre
2	DI2	LT1.2	negre
3	DI3	LT1.3	negre
4	DI4	LT1.4	negre
5	DI5	LT1.5	negre
6	DI6	On/Off	negre
7	DI7	START	negre
8	DI8	Auto/Man	negre
9	DI9	STOP	negre
10	DI10	PE	negre
11	DI11	FT1.1	marró

Taula 15 - Taula de connexions del mòdul X3.

MÒDUL X3 CPU X20 CP 1381			
PIN	Sigles	CONNEXIÓ	CABLE
1	DO1	V1.1	vermell
2	DO2	V1.2	vermell
3	DO3	V1.3	vermell
4	DO4	V1.4	vermell
5	DO5	LLUM POWER	marró
7	DO7	LLUM ALARM	marró
13	(+24V) CPU/X2XL	Alimentació font	vermell
14	(+24V) I/O	Alimentació I/O	blau clar
15	GND	GND CPU	blau
16	GND	GND I/O	blau

Taula 16 - Taula de connexions del mòdul X4.

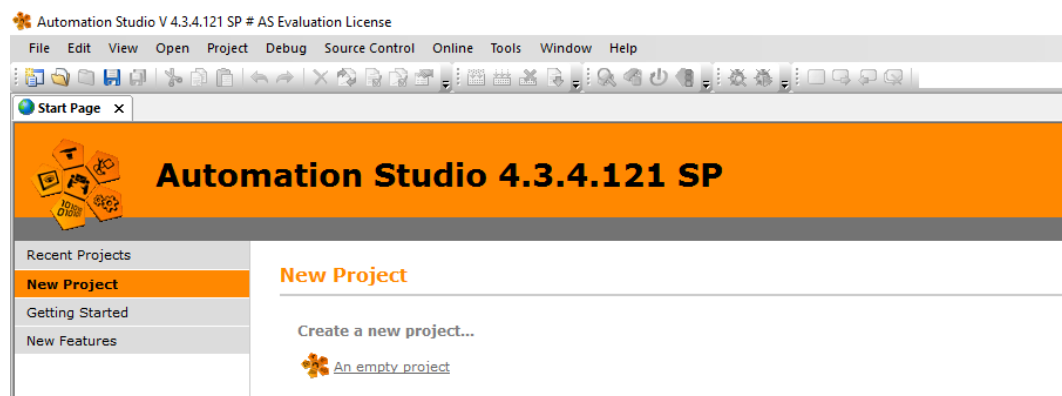
MÒDUL X4 CPU X20 CP 1381			
PIN	Sigles	CONNEXIÓ	CABLE
3	AO+1U	(senyal +) V1.5	negre
4	AO+2U	(senyal +) P1.1	blau
5	AO-1U/I	(-) V1.5	blau
6	AO-2U/I	(-) P1.1	blau

5. ENTORN DE PROGRAMACIÓ

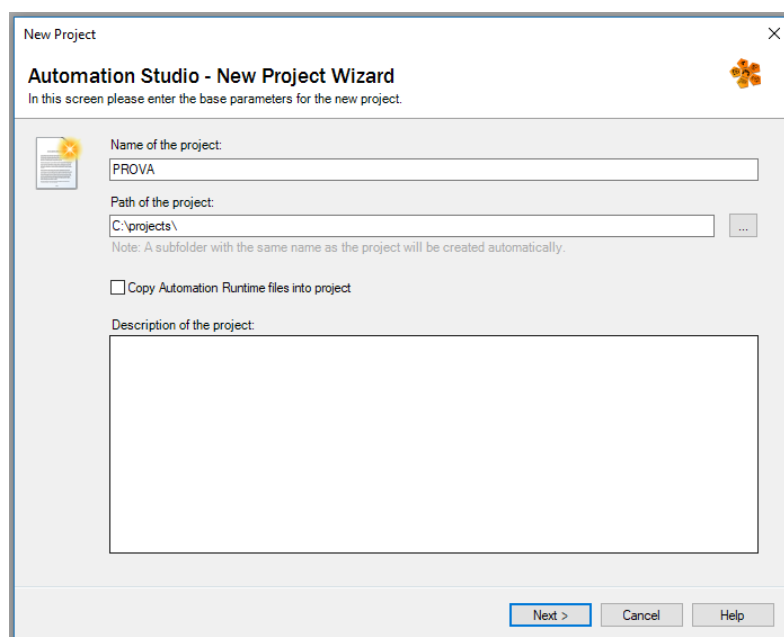
5.1. Nou projecte en Automation Studio

L'entorn de programació d'Automation Studio permet el control, la comunicació i la visualització d'un programa amb diversos llenguatges.

Per començar a utilitzar-lo, primerament obrirem un projecte nou que es podrà fer des de la pagina d'inici ("Start Page" mostrada en la imatge 44) o bé accedint a "file → New project" (imatge 44). S'obrirà una finestra "New Project" que ens permetrà nomenar el projecte i el directori de destí del projecte, podent incorporar una descripció del projecte (imatge 45). Per continuar es prem el botó "Next".

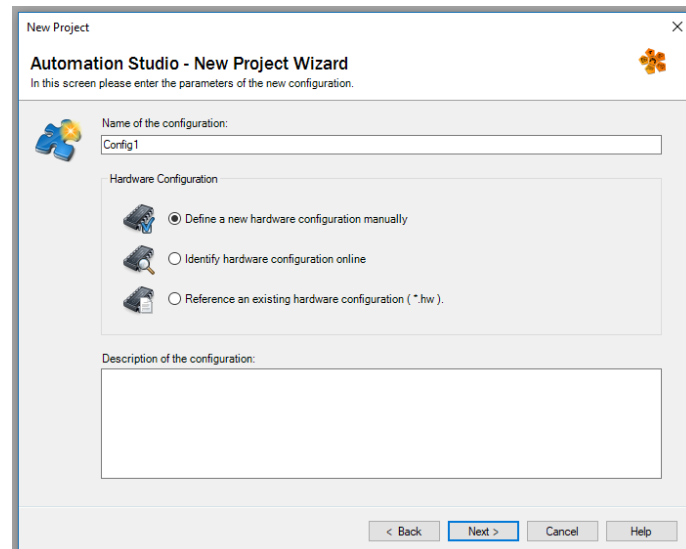


Imatge 44 – Pantalla d'inici per crear un nou projecte en Automation Studio.



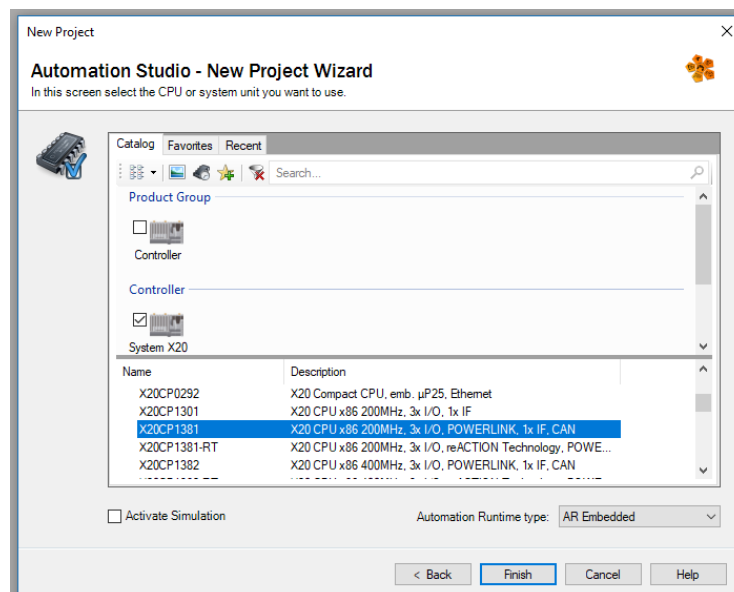
Imatge 45 - Finestra de creació d'un projecte nou en Automation Studio.

Seguidament ens apareixerà una pantalla on podrem nomenar la configuració inicial i reconèixer el hardware de 3 maneres: definir un nou hardware manualment, identificar un hardware online o agafar una configuració existent del hardware amb extensió “.hw” (imatge 46). Un cop triat un dels 3 mètodes de reconeixement es prem el botó “Next” per continuar.



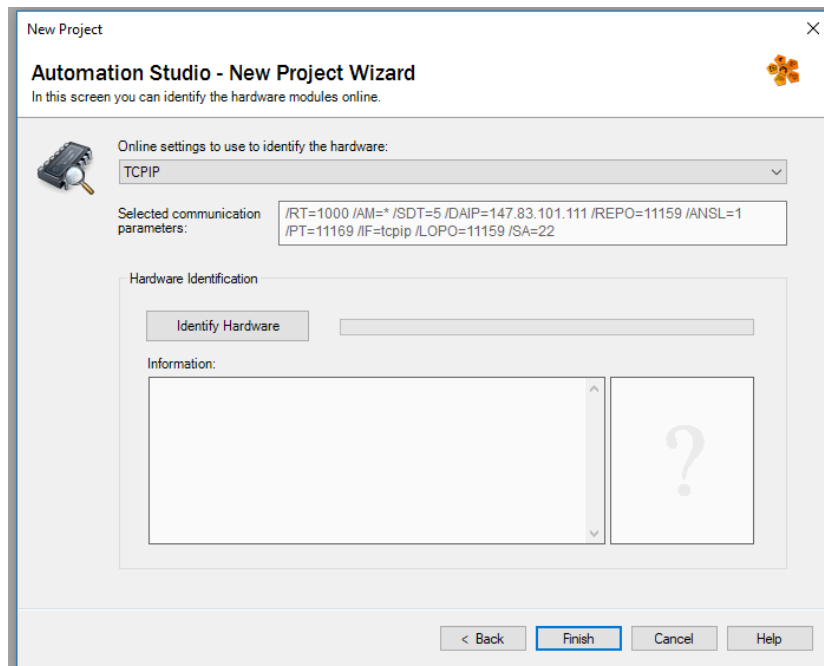
Imatge 46 - Finestra de configuració d'un projecte nou en Automation Studio.

En el cas de seleccionar el reconeixement manual (primera opció), s'obrirà un catàleg on es selecciona la CPU o la unitat del sistema a controlar, per exemple en aquest cas s'incorpora el controlador “System X20” i selecciona el PLC X20CP1381. Un cop seleccionat es prem el botó de “Finish” (imatge 47).



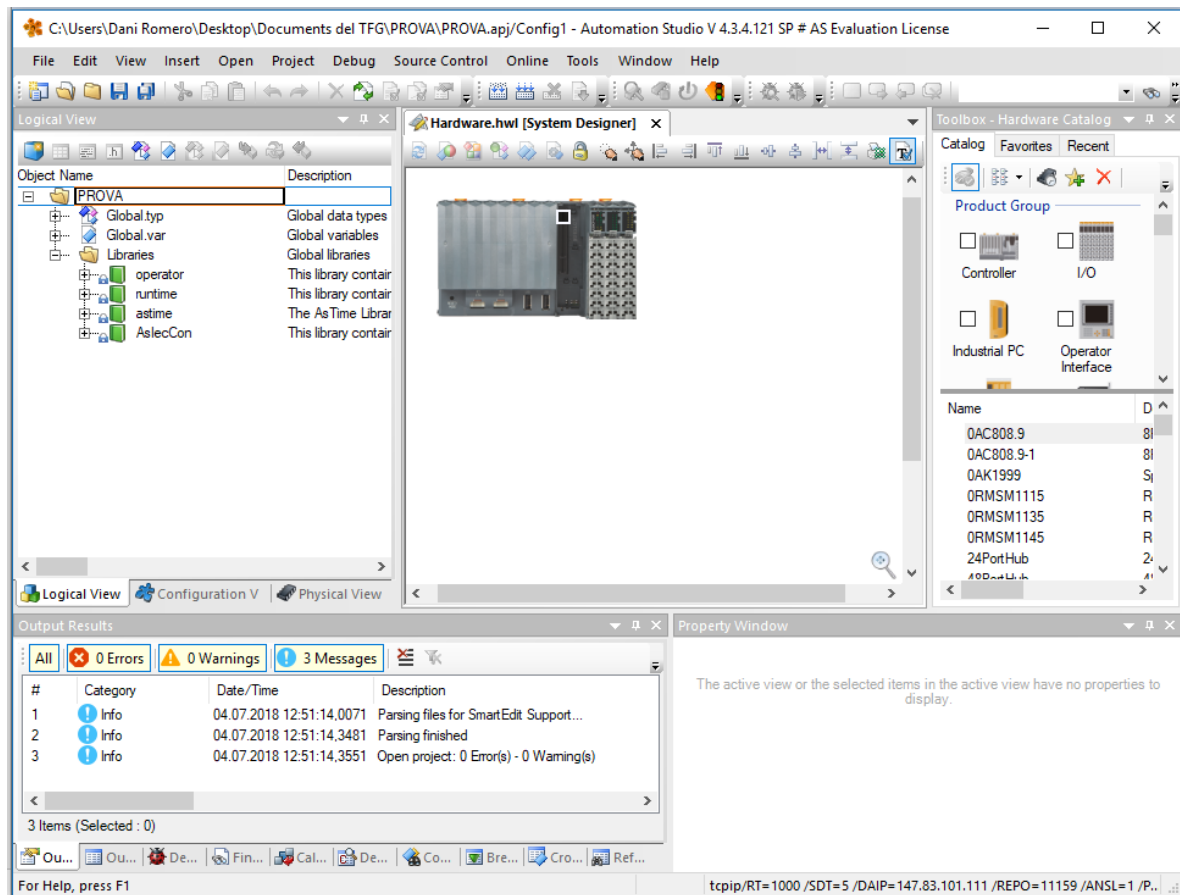
Imatge 47 - Finestra d'identificació de hardware manual d'un projecte nou en Automation Studio.

En cas de seleccionar el reconeixement d'un hardware online s'obrirà una altra finestra on haurem de seleccionar la configuració online per identificar el hardware que pot ser mitjançant una interfície TCPIP, ARsim_TCPIP, ARwin, Serial o mòdem, podent modificar els paràmetres de comunicació. Un cop seleccionada la configuració es prem el botó de "Identify Hardware" i un cop ens pareixi el software es prem el botó de "Finish" (imatge 48).



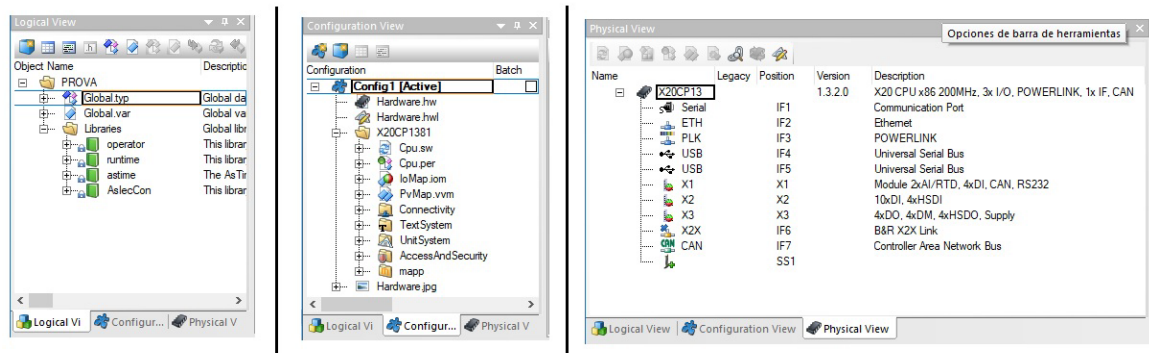
Imatge 48 - Finestra d'identificació automàtica de hardware d'un projecte nou en Automation Studio.

Un cop creat i configurat el projecte apareixerà l'entorn de treball amb diferents finestres (imatge 49):



Imatge 49 - Entorn de treball d'Automation Studio.

- Una finestra en la part esquerra amb 3 opcions a la base d'aquesta: “Logical View” que mostra els programes i l'arxiu on es guarden les variables locals (per variables i funcions que es fan servir tots els programes), així també com algunes llibreries referents a l'estàndard IEC 61131-3; “Configuration View” per configurar el hardware connectat i configurar les connexions entre aquests; i “Physical View” per modificar paràmetres del PLC (I/O, Ethernet, USB, CAN, POWERLINK...) (imatge 50).

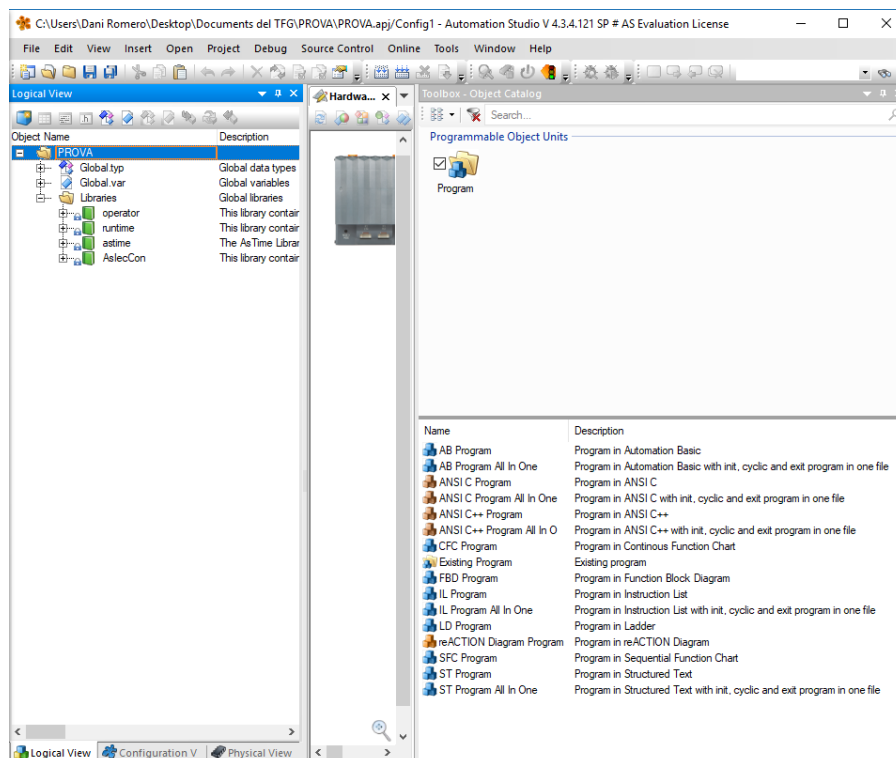


Imatge 50 - Finestra de “Logical, Configuration i Physical View” d'Automation Studio.

- L'àrea de treball és la finestra central que permet modificar el codi font.
- A la finestra superior dreta hi apareix el catàleg d'objectes "Toolbox catalog", segons l'objecte seleccionat apareixeran diferents opcions de hardware, funcions de programa i/o objectes de software.
- La finestra inferior esquerra apareix "output window", mostra la informació que es genera durant la compilació del projecte.
- La finestra inferior dreta incorpora la finestra "Properties window", mostra les opcions de configuració de l'objecte o mòdul de hardware seleccionat.

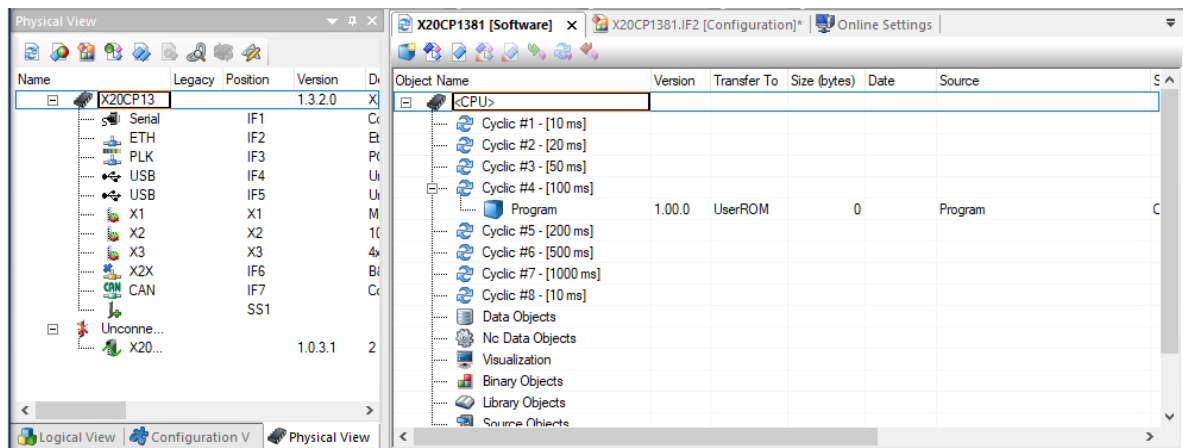
Si no apareixen aquestes finestres, una altra manera de cridar-les és entrant en el menú "View → Go To".

Per crear un programa nou, anem a "Logical View" i seleccionem la carpeta del projecte. Tot seguit anem a "Toolbox" i marquem la casella de "Program" dins de l'apartat "Programmable Object Units". A la part inferior de la finestra apareixen diversos llenguatges de programació. Automation Studio disposa d'un ampli repertori de llenguatges de programació: Automation Basic, ANSI C, ANSI C++, Continuous Function Chart, Bloc Diagram, Instruction List, Ladder, reAcTION Diagram, Sequential Function Chart i Structured Text. Alguns llenguatges incorporen l'opció "All In One", en aquest cas les diferents rutines: inici de programa, final de programa, cyclic, etc., formaran part del mateix programa (no es separen en subprogrames). Per agregar qualsevol d'ells al projecte cal arrossegar-les fins a la carpeta de projecte (imatge 51).



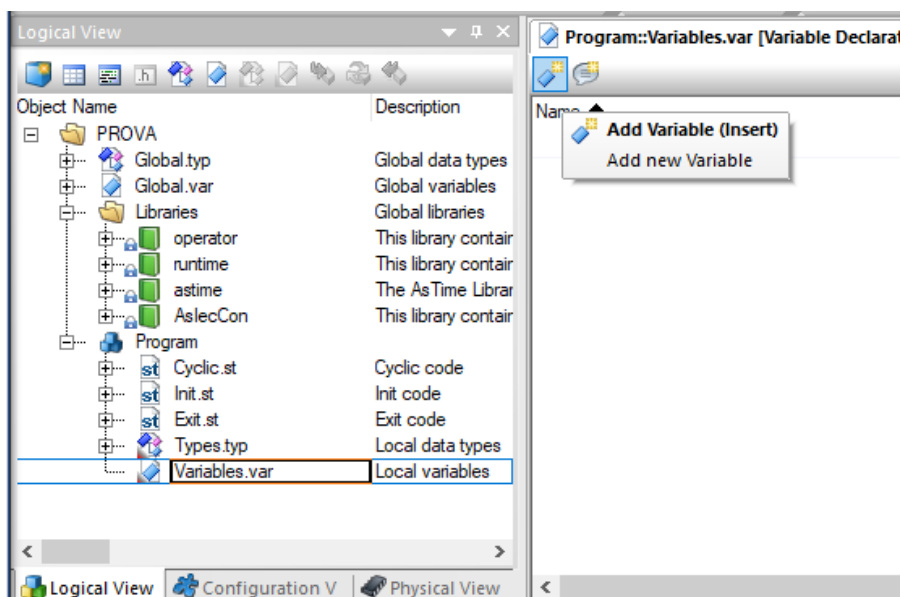
Imatge 51 - Insertar un objecte de programació en Automation Studio.

Sempre que hi hagi programes creats s'hauran d'incloure en "Cyclic" de la CPU que es pot trobar fent doble clic al símbol de la CPU en "Physical View". El "Cyclic" és allà on s'assigna el temps d'execució del programa. Els programes amb temps d'execució més baix tindran prioritat sobre els que tenen un temps d'execució més alt. Els temps d'execució de programa o les velocitats dels "Cyclics" poden ser de l'ordre de 10, 20, 50, 100, 200, 500 o 1000 mil·lisegons. Per assignar un programa a un "Cyclic" només és necessari arrossegar el programa (imatge 52).



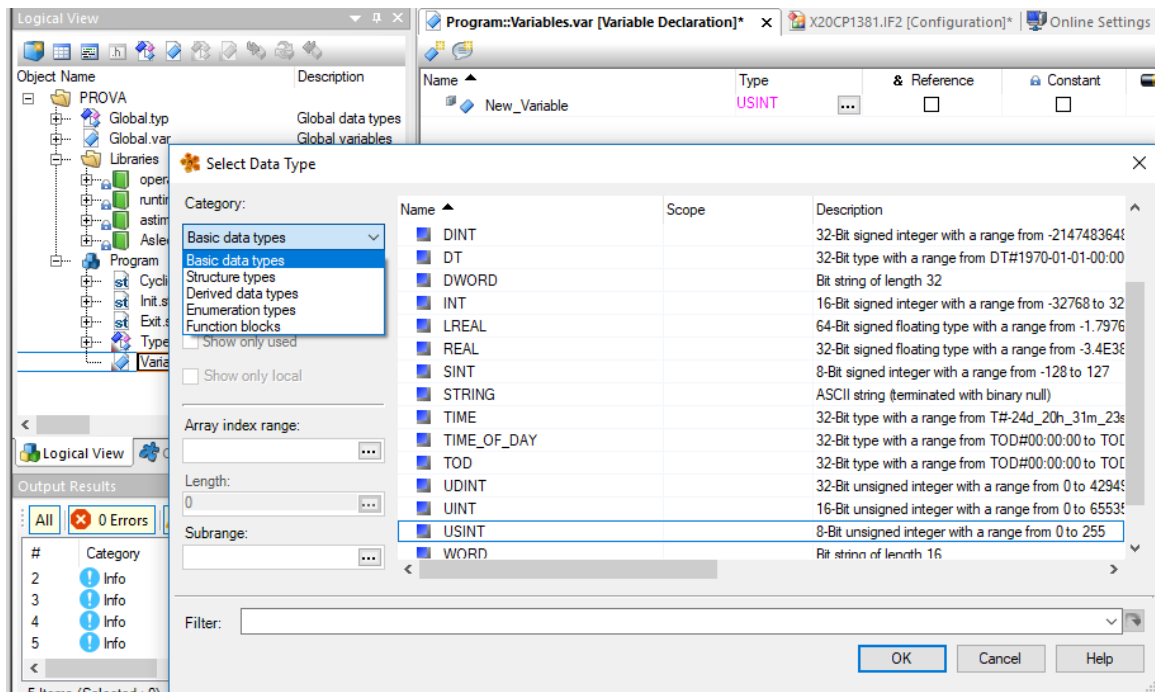
Imatge 52 - Selecció del "Cyclic" del programa en Automation Studio.

Per definir les variables, ja siguin locals o globals, cal anar a la finestra "Logical View" i obrir "Global.var" o "Variables.var" del programa seleccionat. Per afegir variables cal prémer el botó "Add Variable" o bé amb clic dret seleccionar "Add variable" (imatge 53).



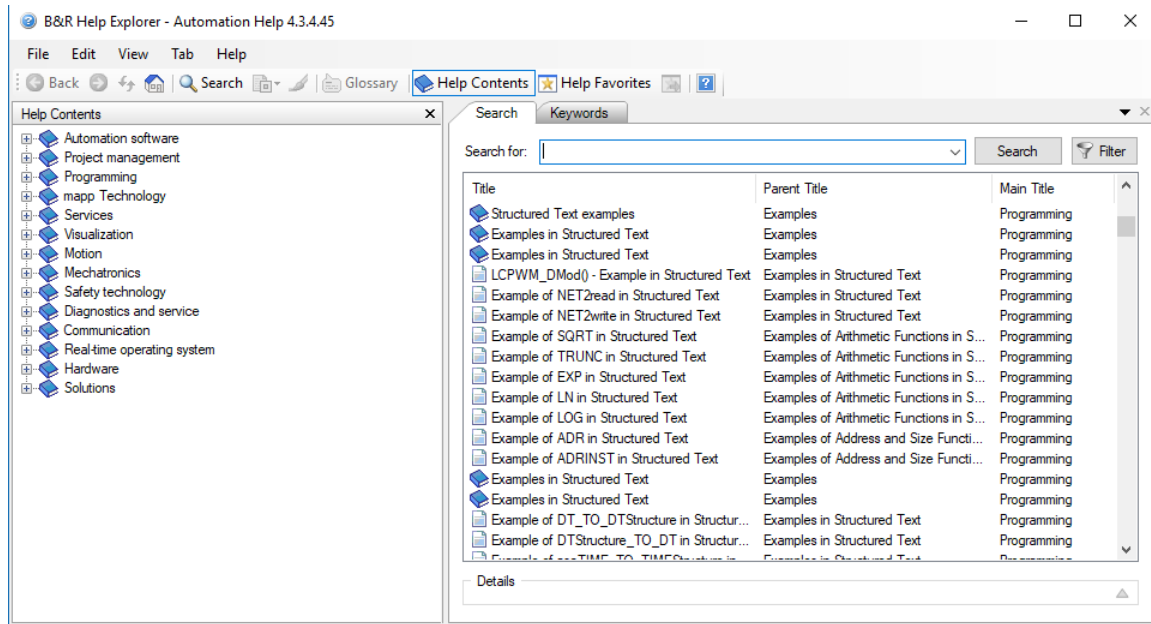
Imatge 53 - Crear una variable en Automation Studio.

Per definir el tipus de variables que es vol, es fa doble clic a l'esquerre a la columna "Type" i es clica el botó "...", apareixent una finestra on podem seleccionar el tipus de variable que es vol filtrant per categories (tipus de dades, tipus d'estructura, tipus de dades derivades, tipus enumerat o bloc de funcions) (imatge 54).



Imatge 54 - Selecció de tipus de variable en Automation Studio.

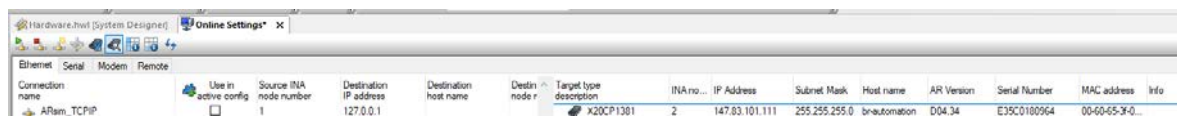
Quan manqui informació sobre un objecte del programa, es pot clicar o bé deixar el punter a sobre aquest objecte i prémer la tecla "F1" del teclat de l'ordinador. Apareix una finestra d'ajuda o "Help Explorer" que té informació de totes les funcions que incorpora Automation Studio. En la part esquerra d'aquesta finestra apareix el contingut d'ajuda o "Help Contents", una subfinestra que conté un buscador ("Search") i la informació de l'objecte o tema sobre el qual s'ha demanat ajuda (imatge 55).



Imatge 55 - Finestra d'ajuda d'Automation Studio.

5.2. Connexió amb el PLC

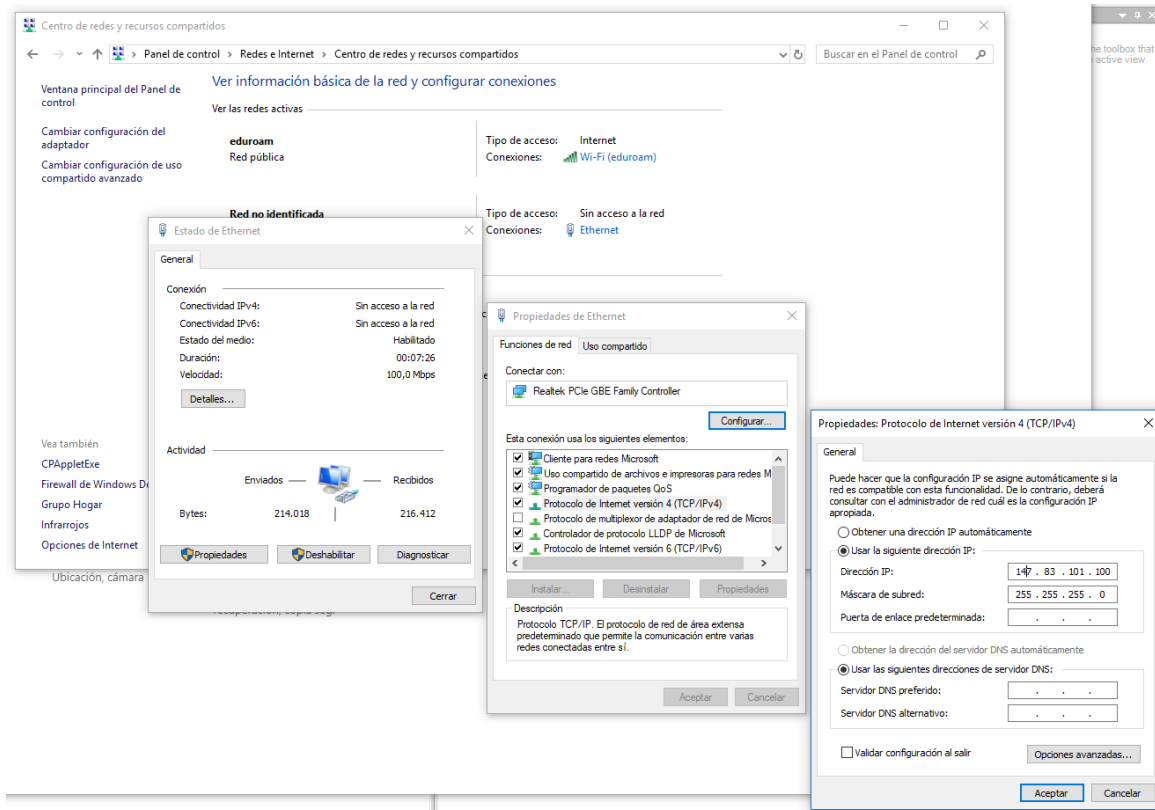
Per connectar-se amb el PLC, cal connectar un cable Ethernet del PLC a l'ordinador i seguidament buscar la direcció IP del controlador. Per saber la direcció IP del controlador s'accedeix al menú “Online → Settings...” i apareix la finestra “Online Settings” amb 3 petites subfinestres en la part superior de la finestra, en aquest cas es selecciona “Ethernet”. Des de aquesta finestra es selecciona l'opció de “Browse” (🔍) i sortirà en la part dreta de la finestra el PLC amb els seus parametres de connexió (imatge 56).



Imatge 56 - Finestra "Online Settings" amb el PLC trobat (Automation Studio).

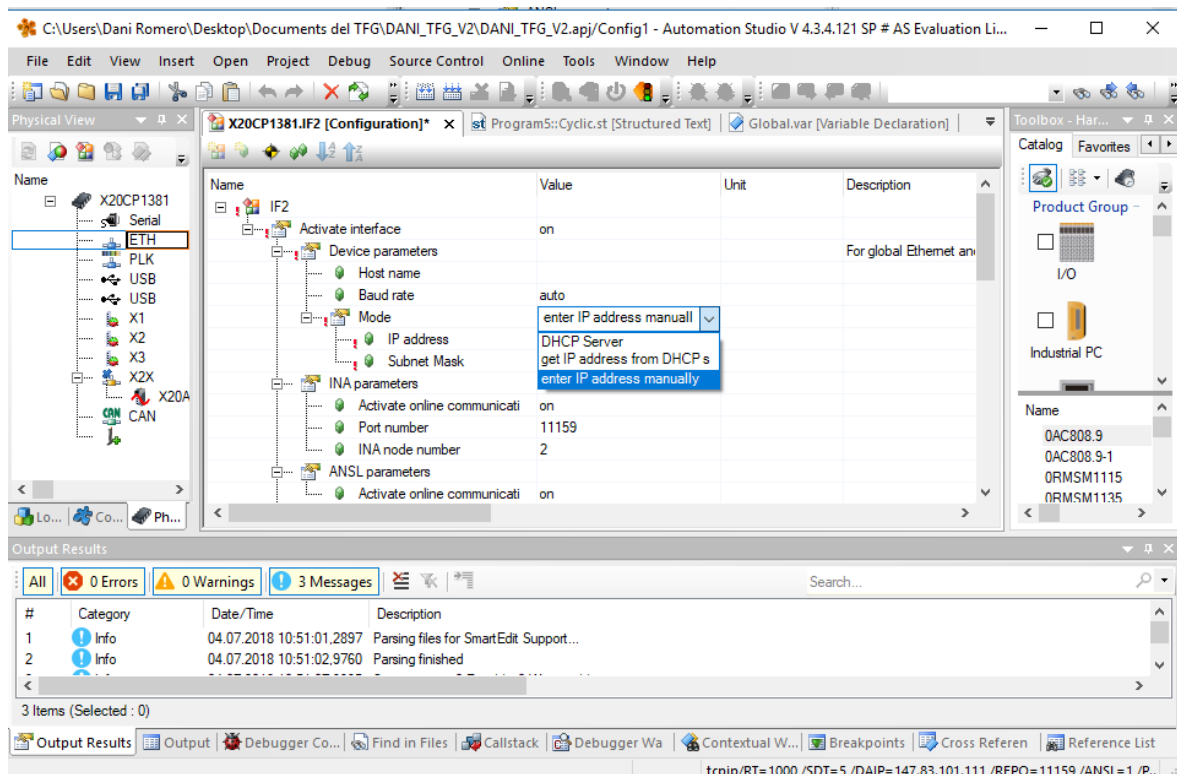
Per connectar el PLC amb el nostre ordinador aquest hauran de coincidir en màscara i els tres primers números de la IP. Per canviar la IP de l'ordinador (guia per Sistema Operatiu Windows) s'accedeix a “Panel de control → Redes e Internet → Centro de redes y recursos”, i en l'apartat “Ver las redes activadas” seleccionem “Ethernet”. S'obrirà una petita finestra que ens dirà l'estat del Ethernet, accedim a “Propiedades”. Dins de la finestra “Propiedades de Ethernet” es selecciona “Protocolo de Internet versión 4 (TCP/IPv4)” i s'accedeix a “Propiedades” i s'habilita la casella de “Usar la siguiente dirección IP”. Seguidament, s'entra l'adreça IP del PLC, aquesta ha de coincidir amb la del PLC en els

seus tres primers números i cal posar la mateixa adreça de la màscara de subxarxa (imatge 57).





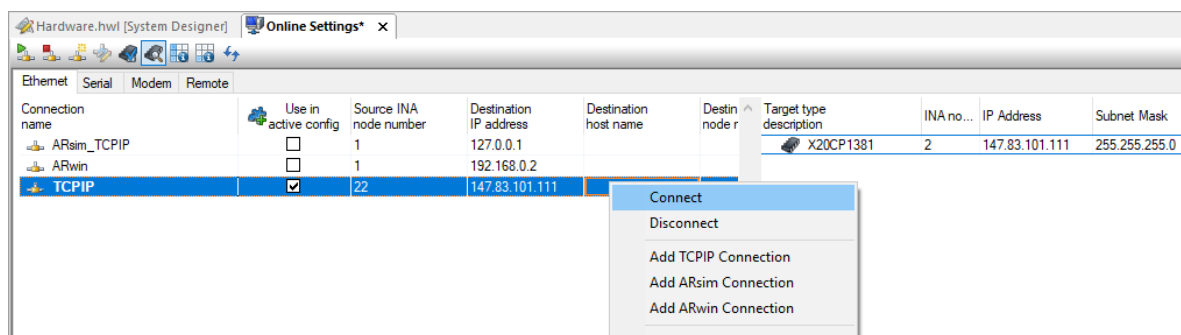
Imatge 57 - Canviar la IP de la connexió Ethernet del PC.

S'accedeix de nou al programa i dins de la finestra "Physical View", es fa clic dret a "ETH" i anem a "Configuration". S'obrirà una nova finestra amb el nom del PLC i amb "[Configuration]*", aquí s'accedeix a "Activate interface → Device paràmetres → Mode" obrint-se un desplegable on es selecciona l'opció "enter IP address manually" on caldrà escriure l'adreça IP (en un rang diferent a l'ordinador i al PLC) i l'adreça de la mascara de la subxarxa (imatge 58).



Imatge 58 - Canviar la IP de la connexió Ethernet en Automation Studio.

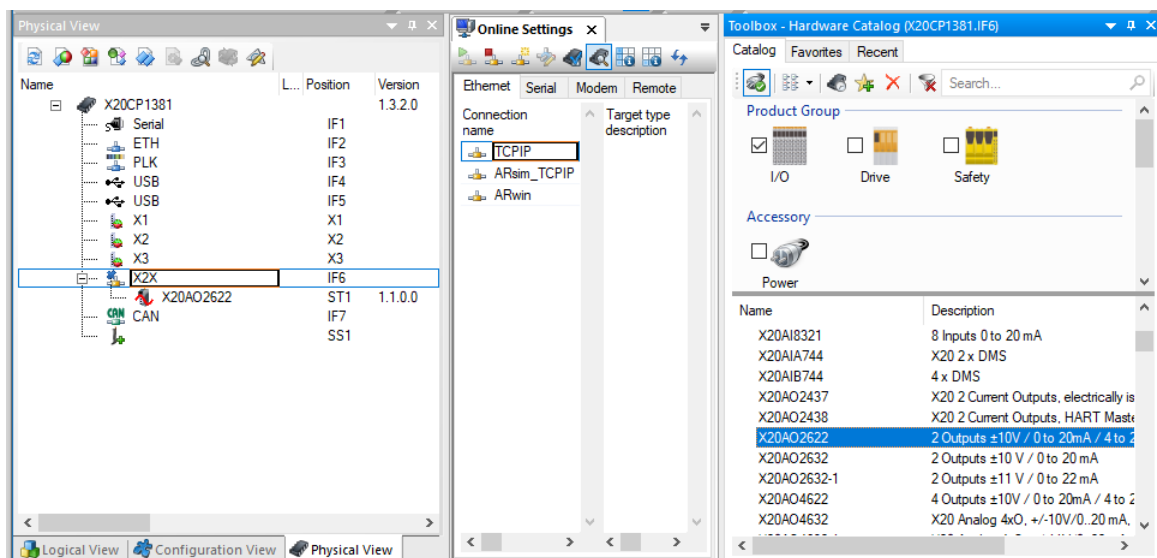
Guardem tots els canvis fets amb la icona “” i tornem a la finestra “Online Settings” i es selecciona l’opció de “Refresh” () per refrescar el dispositiu. En la mateixa finestra, on apareix la connexió “TCPIP” revisem que tingui la mateixa adreça IP de quan s’ha configurat la connexió Ethernet, sinó és així, es torna a escriure. Per connectar-se només falta canviar el número “Source INA node number” i fent clic dret a la connexió “TCPIP” i seleccionant l’opció “Connect” a la part inferior dreta de la finestra del programa es mostrarà “**RUN**” fent referència al fet que ja s’ha connectat amb el PLC (imatge 59).



Imatge 59 - Connectar-se al PLC un cop ja configurades les IP en Automation Studio.

5.3. Incorporar nou hardware

S'accedeix a "Physical View" i, seleccionant la CPU que apareix, anem a "Toolbox" i busquem dins de "Catalog" l'objecte que volem afegir, per exemple el mòdul extra X4 amb referència X20AO2622. Es busca "Product Group" i es selecciona "I/O", dins de l'opció busquem la referència del mòdul i l'arrosseguem a la finestra de "Physical View" on apareixerà el mòdul (imatge 60).



Imatge 60 - Incorporació del mòdul X4 (X20AO2622) en Automation Studio.

5.4. Programació amb Text Estructurat

El Text Estructurat (Structured Text - ST) és un dels cinc llenguatges suportats per l'estàndard IEC 61131-3. És un llenguatge d'alt nivell que està estructurat en blocs: "Init" és la rutina d'inici de programa, "Cyclic" és la rutina d'execució normal del programa, "Exit" és la rutina de sortida de programa.

Les sentències i operadors que té el llenguatge de programació "ST" apareixen en les taules 17 i 18:

Taula 17 - Taula de paraules clau en Text Estructurat.

Keyword	Meaning
ACCESS	Defines dynamic access. See also " Dynamic variables ".
BIT_CLR	A = BIT_CLR(IN, POS) A contains the value IN after the bit at position POS is deleted. However, the IN operand remains unchanged.
BIT_SET	A = BIT_SET(IN, POS) A contains the value IN after the bit at position POS is set. However, the IN operand remains unchanged.
BIT_TST	Determines the value of a bit: A := BIT_TST(IN, POS) A contains the value of the bit at position POS of operand IN.
BY	See FOR statement .
CASE	See CASE statement .
DO	See WHILE statement .
EDGE	Detects positive and negative edges. A frequent application error is skipping an expression with EDGE (i.e. with an IF statement).
EDGENEG	Detects negative edges. A frequent application error is skipping an expression with EDGENEG (i.e. with an IF statement).
EDGEPOS	Detects positive edges. A frequent application error is skipping an expression with EDGEPOS (i.e. with an IF statement).
ELSE	See IF statement .
ELSIF	See IF statement .
END_CASE	See CASE statement .
END_FOR	See FOR statement .
END_IF	See IF statement .
END_REPEAT	See REPEAT statement .
END_WHILE	See WHILE statement .
EXIT	See EXIT statement .
FOR	See FOR statement .
IF	See IF statement .
OF	See CASE statement .
REPEAT	See REPEAT statement .
RETURN	See RETURN statement .
THEN	See IF statement .
TO	See FOR statement .
UNTIL	See REPEAT statement .
WHILE	See WHILE statement .

Taula 18 - Taula d'operadors en Text Estructurat.

Operator	Function
ABS	Returns the absolute value of a number. ABS(-2) returns 2.
ACOS	Returns the arc cosine (inverse function of cosine) of a number.
ADR	Returns a variable's address.
ADRINST	Can only be called from within a function block and returns the instance address of the current function block
AND	Logical AND operation by bit.
ASIN	Returns the arc sine of a number (inverse function of sine).
ASR	Mathematically shifts an operand to the right: A := ASR (IN, N); Shifts IN to the right by N bits and fills in the left with the sign bit.
ATAN	Returns the arc tangent of a number (inverse function of tangent).
COS	Returns the cosine of a number.
EXP	Exponential function: A := EXP (IN).
EXPT	One operand raised to the power of another operand: A := EXPT (IN1, IN2).
LIMIT	Limitation: A := LIMIT (MIN, IN, MAX); MIN is the lower limit, MAX is the upper limit for the result. If IN is less than MIN, then the MIN result is returned. If IN is greater than MAX, then the MAX result is returned. Otherwise, the IN result is returned.
LN	Returns the natural logarithm of a number.
LOG	Returns the base-10 logarithm of a number.
MAX	Returns the largest value as the result.
MIN	Returns the smallest value as the result.
MOD	Modulo division of a USINT, SINT, INT, UINT, UDINT, DINT type variable by another variable of one of these types.
MOVE	The contents of the input variable are copied to the output variable. The := symbol is used as the assignment operator. "A := B;" is the same as "A := MOVE (B);"
MUX	Selection: A = MUX (CHOICE, IN1, IN2, ... INX); CHOICE specifies which of the operators IN1, IN2, ... INX is returned as a result.
NOT	Negation of a bit operand by bit.
OR	Logical OR operation by bit.
ROL	Left bit rotation of an operand: A := ROL (IN, N); IN is shifted N times to the left one bit position at a time, the far left bit being pushed in again from the right.
ROR	Right bit rotation of an operand: A := ROR (IN, N); IN is shifted N times to the right one bit position at a time, the far right bit being pushed in again from the left.

SEL	Binary selection: A := SEL (WAHL, IN1, IN2) CHOICE must be of type BOOL. If CHOICE is FALSE, then IN1 is returned. Otherwise, IN2 is returned.
SHL	Left bit shift of an operand: A := SHL (IN, N); IN is shifted left by N bits and filled from the right with zeros.
SHR	Right bit shift of an operand: A := SHR (IN, N); IN is shifted right by N bits. If IN has a signed data type, the sign bit is used to fill in from the left. If IN has an unsigned data type, zeros are filled in from the left.
SIN	Returns the sine of a number.
SIZEOF	If a variable is specified as the operand, then SIZEOF returns the number of bytes needed by the specified variable. If a type name is specified as the operand, then SIZEOF returns the number of bytes needed by the specified type.
SQRT	Returns the square root of a number.
TAN	Returns the tangent of a number.
TRUNC	Returns the integer part of a number.
XOR	Logical EXCLUSIVE OR operation by bit.

Per declarar un “IF” es fa servir la següent sintaxi:

```
IF <expression1> THEN
  <statement_list1>
ELSIF <expression2> THEN
  <statement_list2>
.
.
ELSIF <expressionN> THEN
  <statement_listN>
ELSE
  <statement_list>
END_IF;
```

Per declarar un “CASE” es fa servir la següent sintaxi:

```
CASE <expression> OF
  <value1> : <statement_list1>
  <value2>, <value3> : <statement_list2>
  <value4>..<value5> : <statement_list3>
  ELSE <statement_list4>
END_CASE;
```

Per declarar un “FOR” es fa servir la següent sintaxi:

```
FOR <control_variable> := <expression1> TO <expression2> BY <expression3> DO  
  <statement_list>  
END_FOR;
```

Per declarar un “REPEAT” es fa servir la següent sintaxi:

```
REPEAT  
  <statement_list>  
UNTIL  
  <expression>  
END_REPEAT;
```

Per declarar un “WHILE” es fa servir la següent sintaxi:

```
WHILE <expression> DO  
  <statement_list>  
END_WHILE;
```

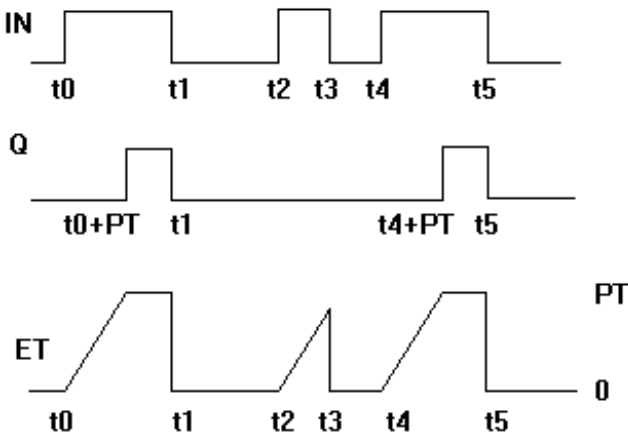
Per declarar blocs de funcions es fa servir la següent sintaxi:

```
<fb_instance> ( <parameter1> := <expression1>, <parameter2> := <expression2>, ...  
<parameterN> := <expressionN> );
```

Per incorporar un temporitzador es fa servir un dels blocs de funcions anomenat “TON”. Aquest temporitzador quan rep el senyal d'entrada “IN” espera un temps “PT” per donar la sortida del temporitzador, on la variable de sortida ve definida per “nom_del_temporitzador.Q”. La sortida es donarà quan el temps transcorregut “ET” (que es pot carregar de la mateixa manera en una variable “nom_del_temporitzador.ET”) sigui igual que el temps d'espera “PT” (taula 19 i imatge61).

Taula 19 - Taula de parametres del temporitzador TON (ST).

I/O	Parameter	Data Type	Description
IN	IN	BOOL	Input signal.
IN	PT	TIME	Delay Time
OUT	Q	BOOL	Output signal, the rising edge of the input signal is delayed by PT.
OUT	ET	TIME	Elapsed time



Imatge 61 - Diagrama de temps de la funció TON.

La declaració del temporitzador









```
TON_01( IN:=input, PT:=T#1s200ms );  
  
output := TON_01.Q;  
  
elapsedTime := TON_01.ET;
```

Per obtenir més informació accedir a l'ajuda d'Automation Studio i en contingut d'ajut accedir a "Programming → Programs → Structured Text (ST)".

5.5. Programació i simulació

Per programar cal compilar el programa per saber si sintàcticament és correcte, l'eina utilitzada és "Build". Si el programa ja s'ha compilat una vegada i es necessita tornar a compilar, hi ha una manera més ràpida que és fent servir l'eina "Rebuild". Per aturar la compilació es fa servir l'eina "Stop Build". Si Automation Studio troba errors sintàctics o bé errors de compatibilitat de variables o altres errors en el programa s'informa en una la finestra inferior "Output Results". Un cop acabada la compilació del programa ens demanarà si volem transferir el programa al PLC, la transferència es farà en cas que no hi hagi errors, que el PLC estigi actiu o en mode simulació. L'eina de "Transfer" transfereix el programa al PLC, compilant el programa abans d'iniciar la transferència.

Si es vol provar el programa sense connectar el PLC, podem activar el mode de simulació mitjançant l'eina "Activate Simulation". En aquest cas, a la part inferior dreta del programa d'Automation Studio s'indica l'estat de la simulació (RUN, OFFLINE, etc.). Per aturar el PLC es fa servir l'eina "Stop Target", en aquest cas s'atura el programa i s'anul·la la connexió amb les entrades i sortides. Per reiniciar el PLC es fa servir l'eina "Warm Restart" (imatge 62).

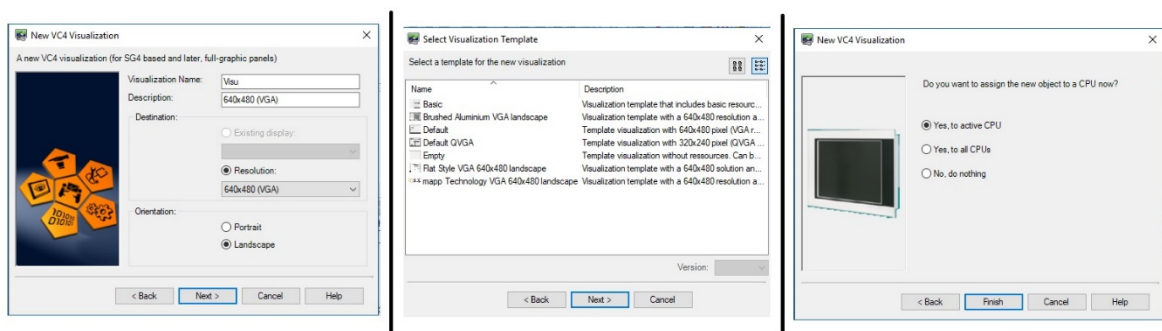
	Build (F7) Builds the entire configuration
	Rebuild (Ctrl+F7) Builds the entire configuration, ignoring dependencies
	Stop Build (Ctrl+Pause) Stops the build of the entire configuration
	Transfer (Ctrl+F5) Transfers the entire configuration to the target
	Monitor (Ctrl+M) Activates monitor mode
	Stop Target Carries out a service restart of the target
	Warm Restart (Ctrl+Mayusculas+W) Carries out a warm restart of the target
	Activate Simulation Activates or deactivates the simulation mode of the target

Imatge 62 - Eines per compilar, transferir, monitoritzar el programa, i eines per controlar i simular el PLC en Automation Studio.

Per observar les variables, l'estat del programa, i els senyals de les sortides i les entrades es fa servir l'eina "Monitor". Aquesta eina obrirà la finestra "Watch" i mostrarà els valors de totes les variables del programa. Fent doble clic sobre el valor d'una variable, entrada o sortida, es pot forçar el valor que es desitgi sempre que sigui compatible.

5.6. Agregar un objecte visualitzador

Per agregar un objecte visualitzador s'accedeix primerament a la finestra "Logical View", seleccionem la carpeta del projecte i tot seguit anem a la finestra "Toolbox". Es busca l'apartat de "Visualization", es marca la casella de "Virtual Components" i arrosseguem l'objecte "VC4 Visualization" (en aquest cas) cap a la carpeta de projecte. Apareix una nova finestra "New VC4 Visualization" que permet nomenar i descriure l'objecte visualitzador, així com definir la resolució i l'orientació del mateix "Portrait / Landscape". Una vegada nomenat i definit, es prem el botó de "Next>" i apareix una finestra que permet triar el format del visualitzador. Aquest pot ser "Basic", "Brushed Aluminium VGA landscape", "Default", "Default QVGA", pantalla buida "Empty", "Flat Styles VGA 640x480" o "mapp Technology VGA 640x480 landscape". Un cop triat el format es prem el botó de "Next" i apareixerà una última finestra on cal seleccionar a quina CPU s'agrega al visualitzador: a la CPU activa, a totes les CPU o a cap. Per acabar premem el botó "Finish" (imatge 63).

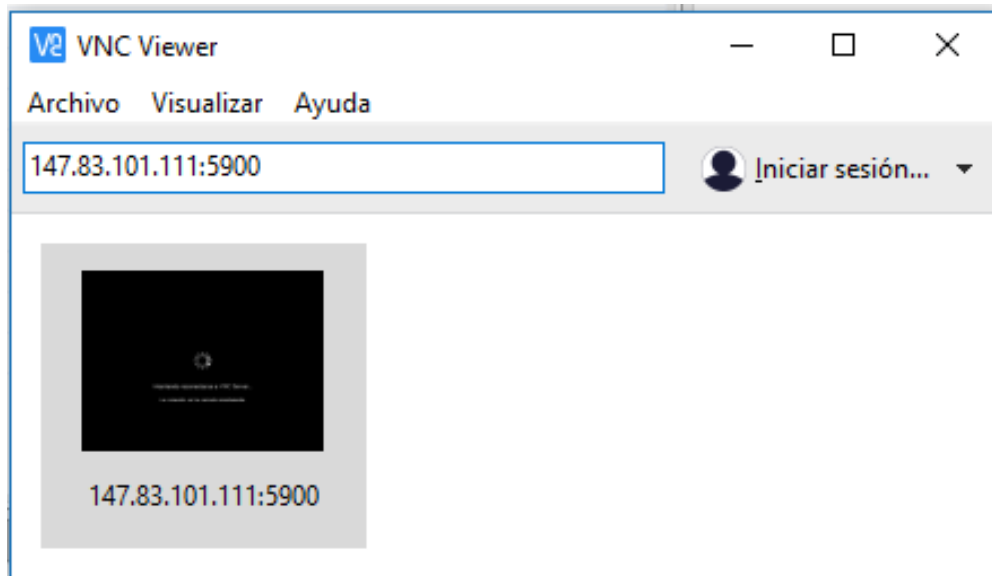


Imatge 63 - Inserció d'un objecte visualitzador en el projecte en Automation Studio.

Per la configuració del teclat d'operacions virtual, s'accedeix al visualitzador creat des de "Logical View" i en l'espai de treball apareixen dues subfinestres: "Visualization" on es presenten les opcions del visualitzador, i "Tools" on apareixen les eines necessàries per configurar el visualitzador amb diferents eines. Per crear una pàgina s'accedeix a "Visualization → Nom_del_visualitzador → Pages → Init. Page" on apareixerà una nova subfinestra "Page: Init_Page" en aquest cas.

Un cop configurat l'objecte visualitzador, podem simular la pantalla fent servir l'aplicació "VNC Viwer". Aquesta aplicació és molt simple de fer servir, únicament val tenir el programa en mode "RUN" ja sigui en mode simulació o connectat al PLC. Primerament s'obre el programa i apareix una finestra amb un buscador on cal especificar una direcció VNC o

una cerca del visualitzador mitjançant el nom. En aquest buscador s'escriu l'adreça VNC configurada i separat per un doble punt, el número de port en un format com per exemple "147.83.101.111 : 5900" i es prem la tecla "Enter". En la finestra de "VNC Viewer" apareixerà el visualitzador que hem especificat en la configuració, fent doble clic sobre aquest s'obrirà el visualitzador amb el teclat d'operacions que s'ha configurat prèviament (imatge 64).



Imatge 64 - Buscar un visualitzador ja configurat en VNC Viewer.

6. AUTOMATITZACIÓ DE L'ESTACIÓ

6.1. Nivells d'automatització

Segons els elements que disposem en la maqueta, es fa un estudi de les possibilitats d'automatització que té l'estació. L'automatització de la maqueta pot estar basada en control de nivell, control de pressió o control de cabal, o incorporant un conjunt d'aquests controls. El nivell d'automatització variarà segons el procés que voldrem dur a terme.

En el TFG es proposen diferents programes, relacionats amb el control de pressió, cadascun amb objectius diferents.

6.2. Automatització de l'estació de treball

L'automatització constarà d'un primer programa que permet recircular l'aigua i detectar fuites durant la seva execució i d'un segon programa, que veurem en l'apartat 6.3, que realitza el control de pressió de l'estació, ambdós programes separats en diferents subprogrames que presenten diferents nivells de complexitat.

Els programes es representaran en diagrames GRAFCET (Gràfic de Control Funcional amb Transició de Etapes, traducció del francès de "Graphe Fonctionnel de Commande Étape Transition"). Són models de representació gràfica dels successius comportaments d'un sistema lògic, predefinit per les seves entrades i sortides. fent més fàcil comprendre el funcionament del programa (Suárez).

6.2.1. Recirculació/Neteja del vas d'expansió

El programa de recirculació d'aigua pel vas d'expansió partirà d'un programa simple que augmentarà en complexitat i en nivell d'automatització permetent adaptar a un usuari amb coneixements bàsics de programació a l'estació de control de nivell, pressió i cabal.

➤ **Programa 1: Graficet de neteja del vas d'expansió**

El primer programa es basa en una recirculació d'aigua del dipòsit inferior cap al vas d'expansió i retornant l'aigua des de la sortida superior del vas d'expansió cap al dipòsit inferior novament. En acabar, el dipòsit inferior retornarà al nivell d'aigua inicial.

Els actuadors que intervenen en el procés són les electrovàlvules V1.1, V1.2, V1.3 i la motobomba P1.1. En el teclat d'operacions es comença per fer servir simplement el polsador de "START" o marxa, encara que tant els selectores d'ON/OFF i l'aturada d'emergència funcionen estan en estat ON, ja que aquets actuen directament com a interruptors en el circuit elèctric.

L'objectiu d'aquest primer programa és que, partint d'un nivell inicial d'aigua qualsevol en el tanc de baix, es bombi l'aigua cap al vas. Prèviament es farà un temporitzador que obrirà les electrovàlvules V1.1 i V1.2 per evitar que en la següent acció la bomba s'activi abans d'obrir-se les electrovàlvules V1.1 i V1.2. Quan es detecta que el vas està ple i l'aigua comença a recircular, es mantindrà durant un temps predefinit. Un cop finalitzat el temps, el vas d'expansió es buida mitjançant la pressió hidroestàtica (pressió deguda al pes d'un fluid en repòs) a causa de la diferència de nivell amb el dipòsit inferior.

El funcionament d'aquest primera automatització constarà de 5 etapes:

E0- En l'etapa 0 el programa estarà esperant el polsador de "START" per començar el cicle.

E1- En l'etapa 1 s'obriran les electrovàlvules V1.1 i V1.2 durant un temps "t1" d'1 segon que proporciona una antelació respecte a l'activació de la bomba, al mateix temps que es memoritza el nivell inicial en el tanc de baix ("nivell A") com a referència, mitjançant el sensor LT1.6.

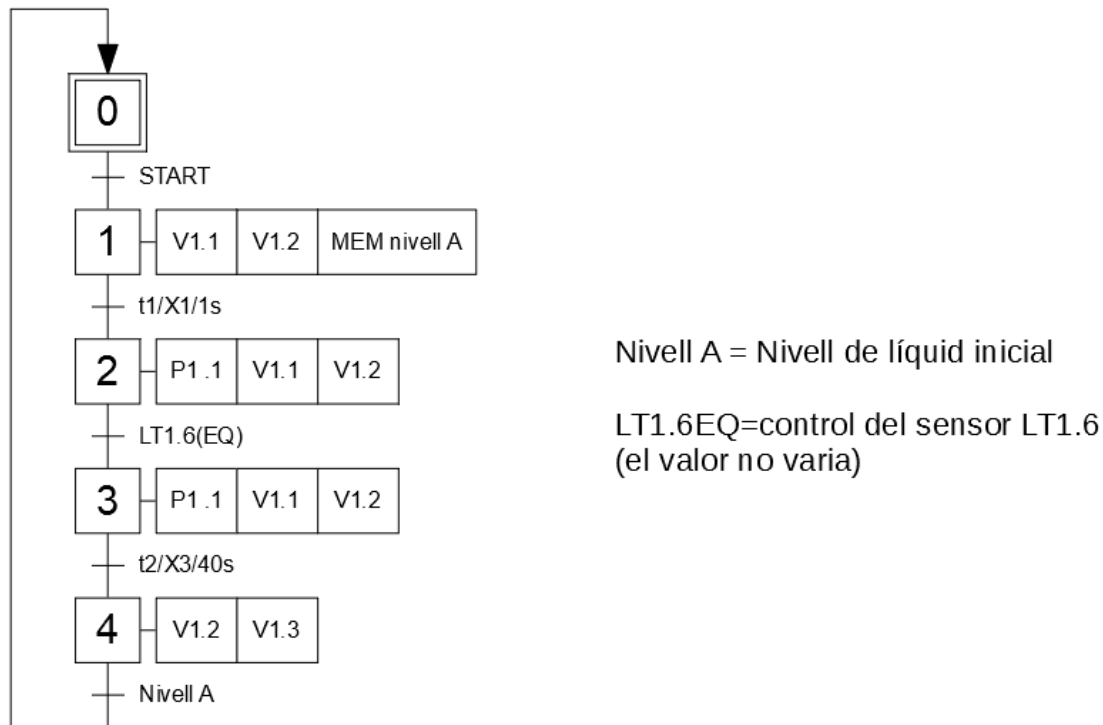
E2- En l'etapa 2 es mantenen obertes les electrovàlvules V1.1 i V1.2 al mateix temps que s'encén la motobomba. El sensor LT1.6 agafa valors en cada instant i fa la mitja de "N" mostres ("LT16A") i la compara amb la mitja de "N" mostres anterior ("LT16A2") amb la finalitat de saber quan l'aigua surt del vas d'expansió i recircula pel dipòsit de baix ("LT16EQ").

E3- En l'etapa 3 intervenen els mateixos actuadors que en l'etapa 2, simplement s'incorpora un temporitzador que indicarà el temps de recirculació de l'aigua pel vas d'expansió ("t2").

E4- En l'última etapa, es para la bomba i només actuen les electrovàlvules V1.2 i V1.3 per retornar al nivell inicial d'aigua que tenien el vas d'expansió i el dipòsit inferior en l'etapa 1.

(NOTA: El caràcter “.” representa una porta lògica “AND”, i el caràcter “+” representa una porta lògica “OR”).

El Grafcet resultant del programa 1 es mostra en la imatge 65.



Imatge 65 - Grafcet del programa 1.

➤ **Programa 2: Grafcet de neteja del vas d'expansió amb seguretat**

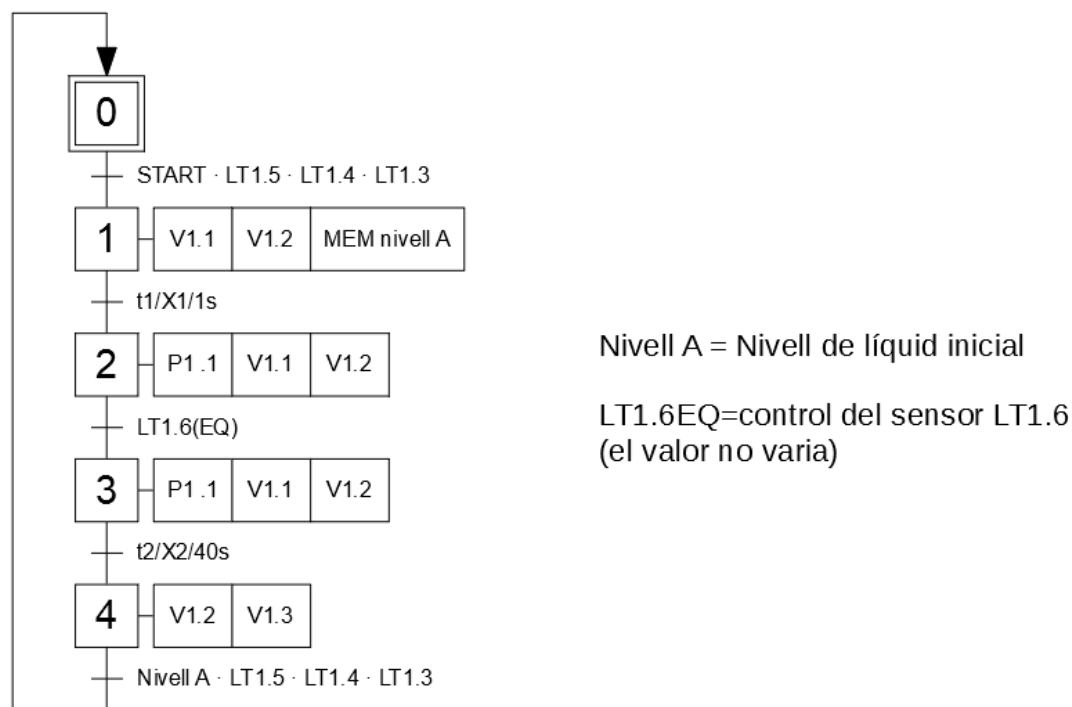
El segon programa té com a objectiu incorporar una seguretat bàsica en el sistema on, en cas que el sensor analògic de nivell falli, mantenir uns límits de seguretat en les etapes inicials i final que permeti garantir el correcte funcionament de l'estació.

Per tant els canvis incorporats en el funcionament del sistema es troben en les següents etapes:

E0- En la transició de l'etapa 0 s'incorpora els sensors capacitius LT1.3, LT1.4 i LT1.5 per saber que s'inicia el cicle amb el dipòsit ple (LT1.5) i que els altres sensors indiquen presència d'aigua en el dipòsit.

E4- En la transició de l'etapa 4 s'incorpora de la mateixa manera els sensors capacitius LT1.3, LT1.4 i LT1.5 per retornar a les condicions inicials.

El Grafcet resultant del programa 2 es mostra en la imatge 66.



Imatge 66 - Grafcet del programa 2.

➤ **Programa 3: Grafcet de neteja del vas d'expansió amb seguretat de límits de nivell**

El tercer programa considera que en el dipòsit inferior pot haver-hi pèrdua d'aigua, ja sigui perquè la clau de pas que es troba en el dipòsit inferior està oberta o perquè una estació esta redireccionant l'aigua del dipòsit inferior de l'estació 1 cap a una altra estació.

Per dur a terme aquest control, en l'etapa on la bomba s'acciona per començar el procés de recircular l'aigua, s'incorpora altres etapes que segons la quantitat d'aigua que hi ha en el dipòsit de dalt en el moment en què es detecta que falta aigua en el dipòsit de baix es realitza una acció o un altre.

Les accions requerides segons els estats esmentats anteriorment s'implementen dues transicions noves de l'etapa 2 i dues noves etapes que contemplin quina acció es realitza segons les condicions de la transició:

(NOTA: Els sensors marcats amb el sufix “*” indiquen que el sensor no està actiu.)

E2- En l'etapa 2 es divideix en tres transicions:

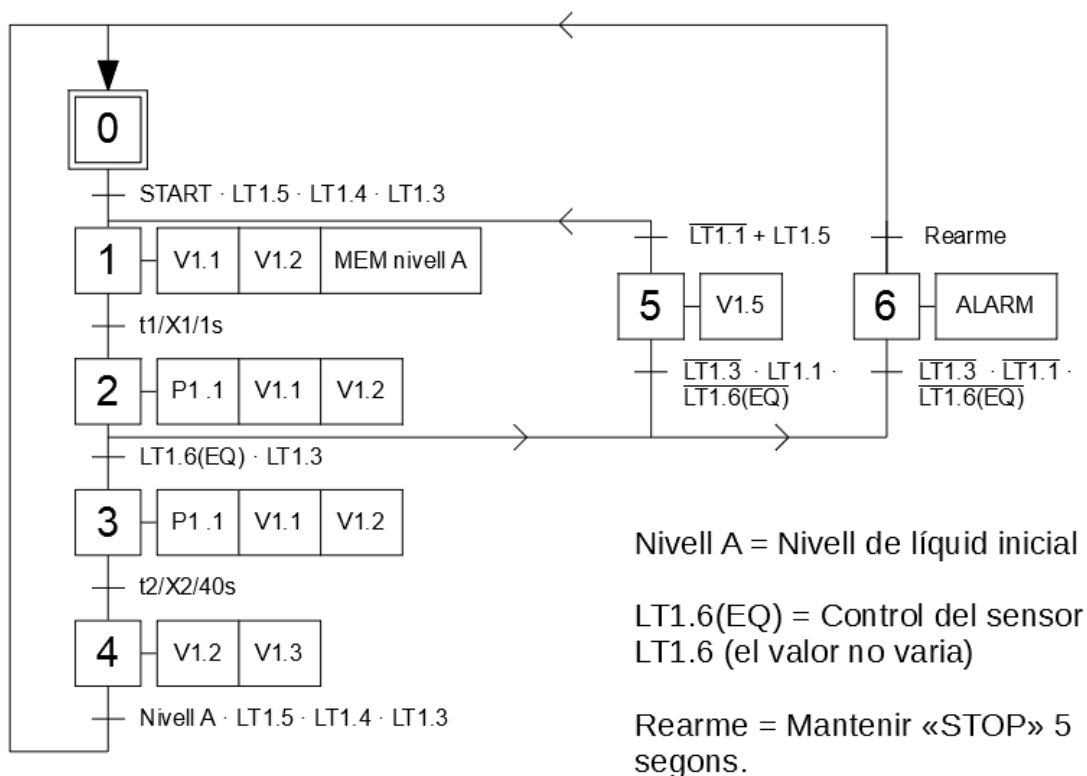
- La transició T2A que activa l'etapa 3 considera quan l'aigua recircula (“LT1.6EQ”) i quan el sensor LT1.3 es troba actiu, és a dir que ens marca nivell mínim, així es pot discriminar entre les tres etapes.

- B. La transició T2B que activa l'etapa 5 considera que no es detecti el sensor LT1.3 (és a dir no ens marca nivell mínim), que no recirculi l'aigua ("LT1.6EQ*") i que es detecti LT1.1 (referent a què hi ha aigua en el dipòsit de dalt), si es compleix salta a l'etapa 5.
- C. La transició T2C que activa l'etapa 6 considera que no es detecti el sensor LT1.3 (és a dir no ens marca nivell mínim), que no recirculi l'aigua ("LT1.6EQ*") i que no es detecti LT1.1 (referent a que no hi ha aigua en el dipòsit de dalt), si es compleix salta a l'etapa 6.

E5- En l'etapa 5 només s'accionarà l'electrovàlvula V1.5, fent que l'aigua del dipòsit superior aboqui l'aigua cap al dipòsit inferior. Un cop es detecti que ja no queda aigua en el dipòsit superior (LT1.1*) o que el dipòsit inferior s'ha omplert fins al límit superior (LT1.5) retorna a l'etapa 1 per tornar a fer el cicle.

E6- En l'etapa 6 només s'acciona la llum de "ALARM" per avisar que els dos dipòsits es troben en els límits inferiors i que cal abocar-hi més aigua, en el dipòsit inferior preferiblement. Per sortir de l'etapa 6 cal accionar el "REARME" que consisteix en mantenir durant cinc segons el pulsador de "STOP" on després retorna a l'etapa 0.

El Grafcet resultant del programa 3 es mostra en la imatge 67.



Imatge 67 - Grafcet del programa 3.

➤ **Programa 4: Grafcet de neteja del vas d'expansió amb seguretat de límits de nivell i control de fuites**

El quart programa contempla que puguin haver-hi fuites durant la recirculació d'aigua pel vas d'expansió, ja sigui per danys en el vas o bé per fuites en les canonades. Si s'arriba a detectar una fuga, s'atura el procés, s'activa l'alarma en mode pampallugues ("ALARM BLINK").

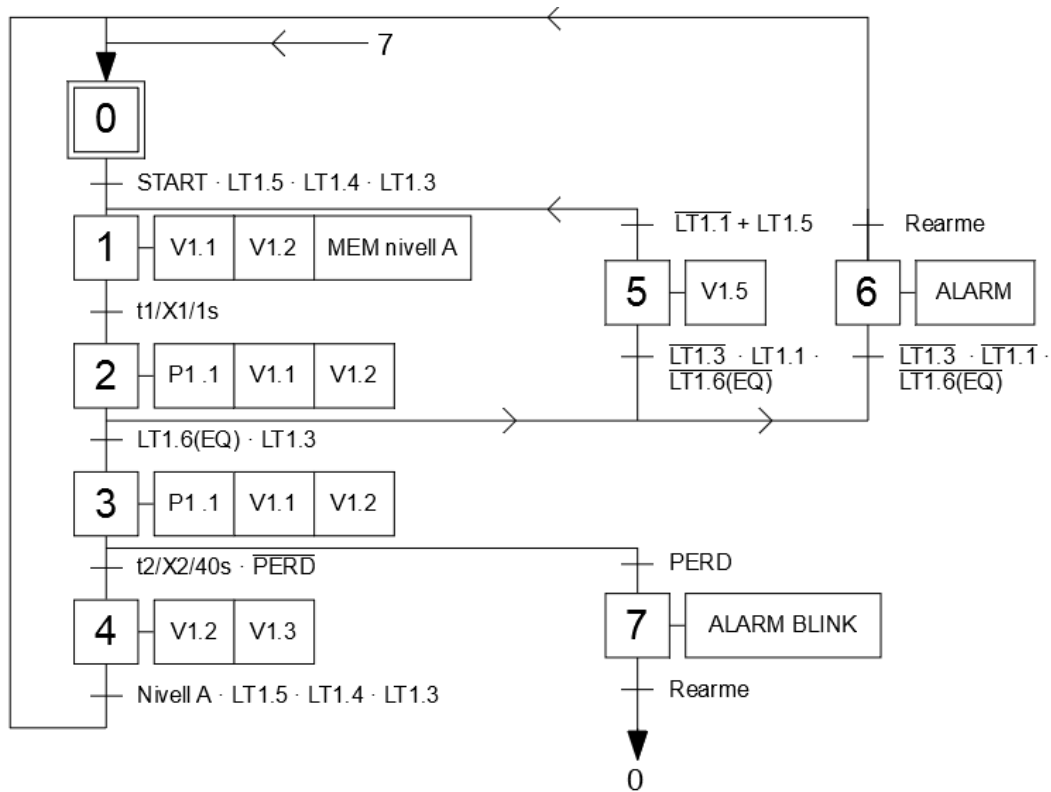
Per implementar aquest sistema cal incorporar una transició més en l'etapa 3 junt amb una nova etapa de l'estat d'alarma en mode pampallugues:

E3- En l'etapa 3 es divideix en dues transicions:

- A. La transició T3A que activa l'etapa 4 considera el temporitzador "t2" i la no activació del senyal virtual "PERD*" (senyal de detecció de fuga), si es compleix salta a l'etapa 4.
- B. La transició T3B que activa l'etapa 7 només considera l'activació del senyal virtual "PERD*" (senyal de detecció de fuga), si és compleix salta a l'etapa 7.

E7- En l'etapa 7 només s'acciona la llum "ALARM" en mode pampallugues avisant que hi ha una fuga durant la recirculació. Per sortir de l'etapa 7 cal accionar el "RE-ARME" que consisteix en mantenir durant cinc segons el pulsador de "STOP" on després retorna a l'etapa 0.

El Grafcet resultant del programa 4 es mostra en la imatge 68.



Nivell A = Nivell de líquid inicial

LT1.6(EQ) = Control del sensor
LT1.6 (el valor no varia)

Rearme = Mantenir «STOP» 5
segons.

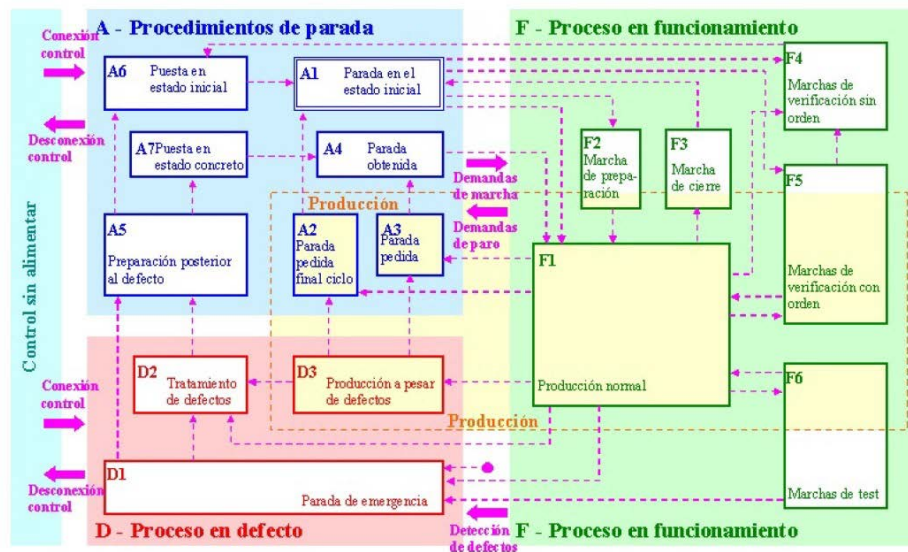
PERD = Detecció de fuga

Imatge 68 - Grafcet del programa 4.

➤ **Programa 5: Grafcet de neteja del vas d'expansió amb seguretat de límits de nivell, control de fugues amb aturada d'emergència i modes manual i automàtic.**

El cinquè programa es pretén fer l'estació 1 completament funcional incorporant els modes d'operació manual i automàtic junt amb la correcta administració del programa, en cas de pulsar l'aturada d'emergència. Per aconseguir aquests objectius s'aplica la guia GEMMA (Guia d'Estudi dels Modes de Marxa i Parada), una guia que contempla els modes de funcionament durant el procés, els procediments d'aturada i posta en marxa i els procediments d'avaría o parades que hi poden haver durant el transcurs del procés (imatge 69).

Modes de marxa i aturada: la guia GEMMA



Imatge 69 - Estats de la guía GEMMA.

Grup A: Procediments d'aturada i posada en marxa

- **A1 Aturada a un estat inicial:** Estat de repòs de la màquina
- **A2 Aturada sol·licitada a un final de cicle:** estat transitori de finalització d'etapes fins a l'estat inicial
- **A3 Aturada a un estat determinat:** estat en que es pot aturar la màquina diferent del inicial
- **A4 Aturada obtinguda:** estat de repòs diferent del inicial
- **A5 Preparació per a la posada en marxa després d'una avaria:** accions a realitzar per retornar la màquina a l'estat de funcionament
- **A6 Portar el sistema a l'estat inicial:** accions a realitzar per reinicialitzar el sistema
- **A7 Portar el sistema a un estat determinat:** retornar el sistema a una posició diferent a la inicial

Grup F: Procés de funcionament

- **F1 Producció normal:** Estat en que la màquina produeix normalment
- **F2 Marxa de preparació:** accions a realitzar perquè la màquina entri en producció (pre-escalfament, ...)
- **F3 Marxa de tancament:** tasques de buidat o neteja
- **F4 Marxa de verificació sense ordre:** control manual, s'utilitza per tasques de verificació i manteniment
- **F5 Marxa de verificació en ordre:** control semi-automàtic, realitza un cicle complet al ritme que determina l'operari
- **F6 Marxa de test:** per realitzar operacions d'ajust i manteniment preventiu

Grup D: Procediments d'avaria

- **D1 Diagnòstic i/o tractament de l'avaria:** estat assolit després d'una aturada d'emergència, cal tenir en compte els procediments d'aturada com les precaucions necessàries per evitar i limitar les conseqüències derivades d'una avaria
- **D2 Aturada sol·licitada a un final de cicle:** en aquest estat la màquina pot ser examinada i, amb o sense l'ajuda de l'operador, indicar els motius de l'avaria per el seu rearmament
- **D3 Producció malgrat les avaries:** casos en què cal continuar produint malgrat la presència d'una avaria, inclou per exemple situacions en que cal finalitzar un reactiu no emmagatzemable, substitució transitòria del treball de la màquina per operari, ...

Per dissenyar la guia GEMMA del procés de recirculació es comença considerant els següents estats: l'Aturada a l'estat inicial (A1) com a punt de partida i els modes de funcionament de producció normal o automàtic (F1) i el mode de marxa de verificació sense ordre o manual (F4). Els procediments d'aturada o avaria que es volen considerar són l'aturada sol·licitada a un final de cicle (A2) en cas d'activar el pulsador de "STOP" i el tractament d'avaria (D1) en cas d'activar l'aturada d'emergència ("PE").

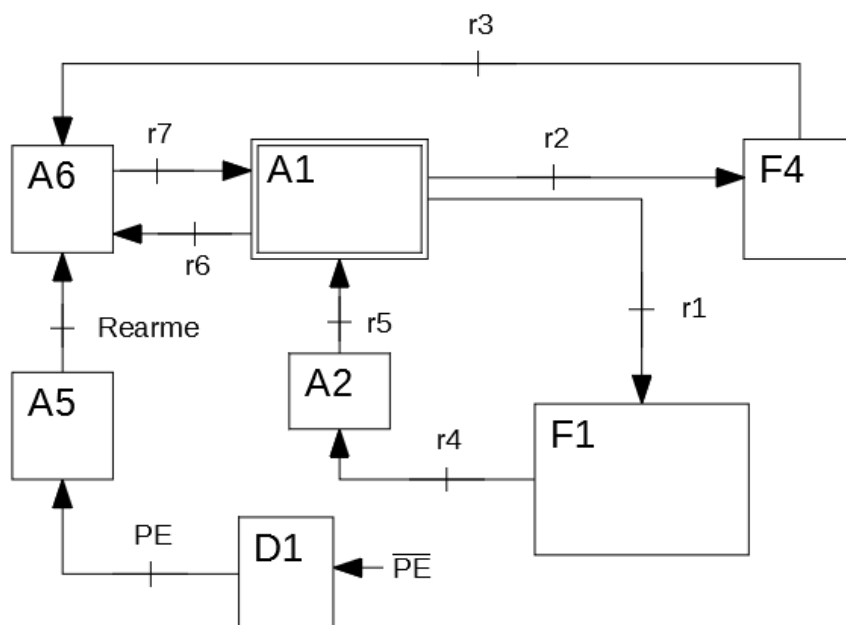
Si premem emergència i el rearmem, retrocedint el bolet d'emergència, o bé per un estat d'alarma o avaria, en aquest cas les etapes E6 i E7, en aquests casos entrem en l'estat de

preparació per a la posada en marxa després d'una avaria (A5) que es trobarà dins del Grafcet d'emergència.

Un cop es surti de l'estat de preparació després d'una avaria (A5) o del mode manual (F4) o que iniciem el sistema per primer cop (A1), pot ser que el sistema no es trobi en l'estat inicial, per tant s'hi apliquen unes transicions cap a l'estat de portar el sistema a l'estat inicial (A6). Quan es porti el sistema a l'estat inicial el sistema es quedarà de nou en l'estat A1.

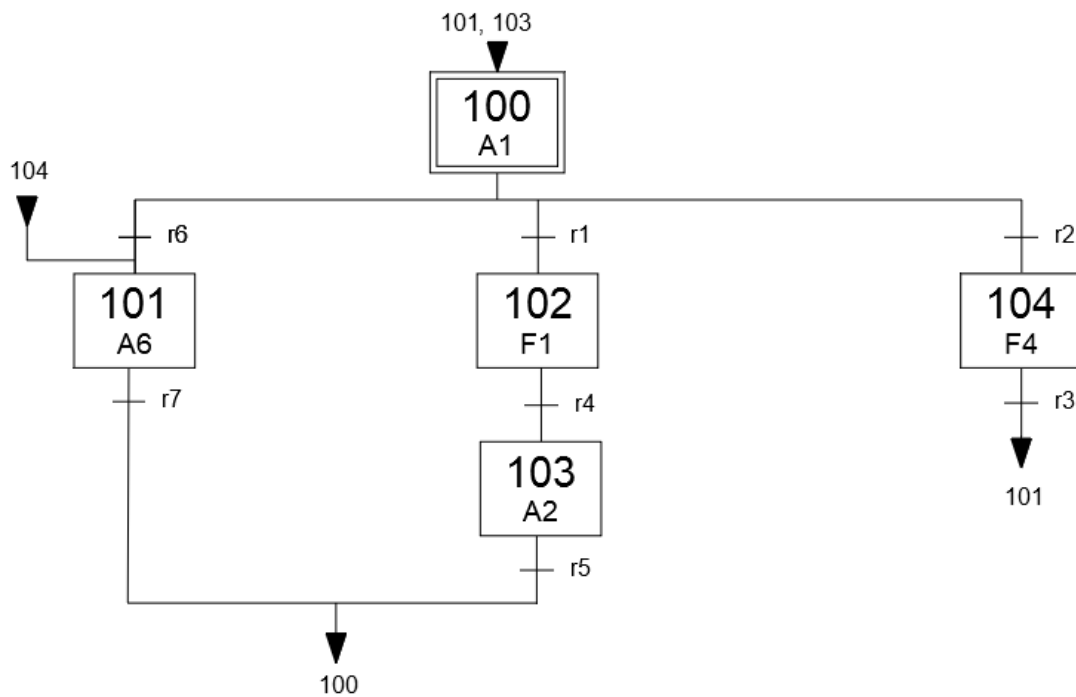
Considerant el procés de recirculació que es porta a terme, el modes que volem incorporar (manual i automàtic) i els tipus d'aturada que es volen implementar (parada el final de cicle), el disseny de la guia GEMMA del programa de neteja del vas d'expansió és el que es mostra en la imatge 70.

GEMMA:



Imatge 70 - Guia GEMMA del programa 5.

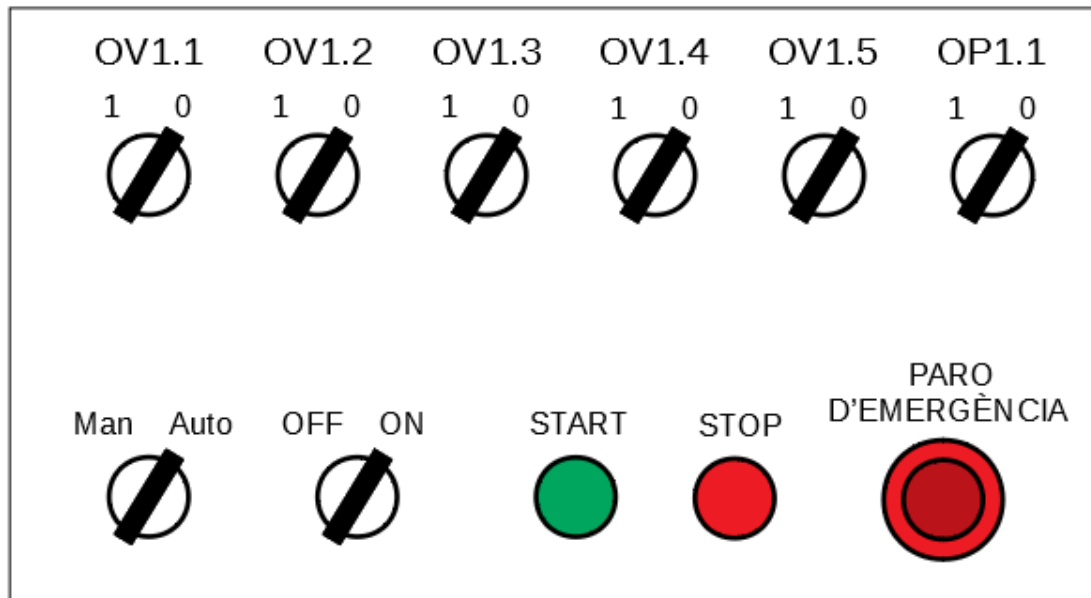
Cal passar la guia GEMMA a Grafcet per entendre el flux d'execució i definir les etapes que activaran els estats del GEMMA (imatge 71).



Imatge 71 - Grafcet de la guia GEMMA del programa 5.

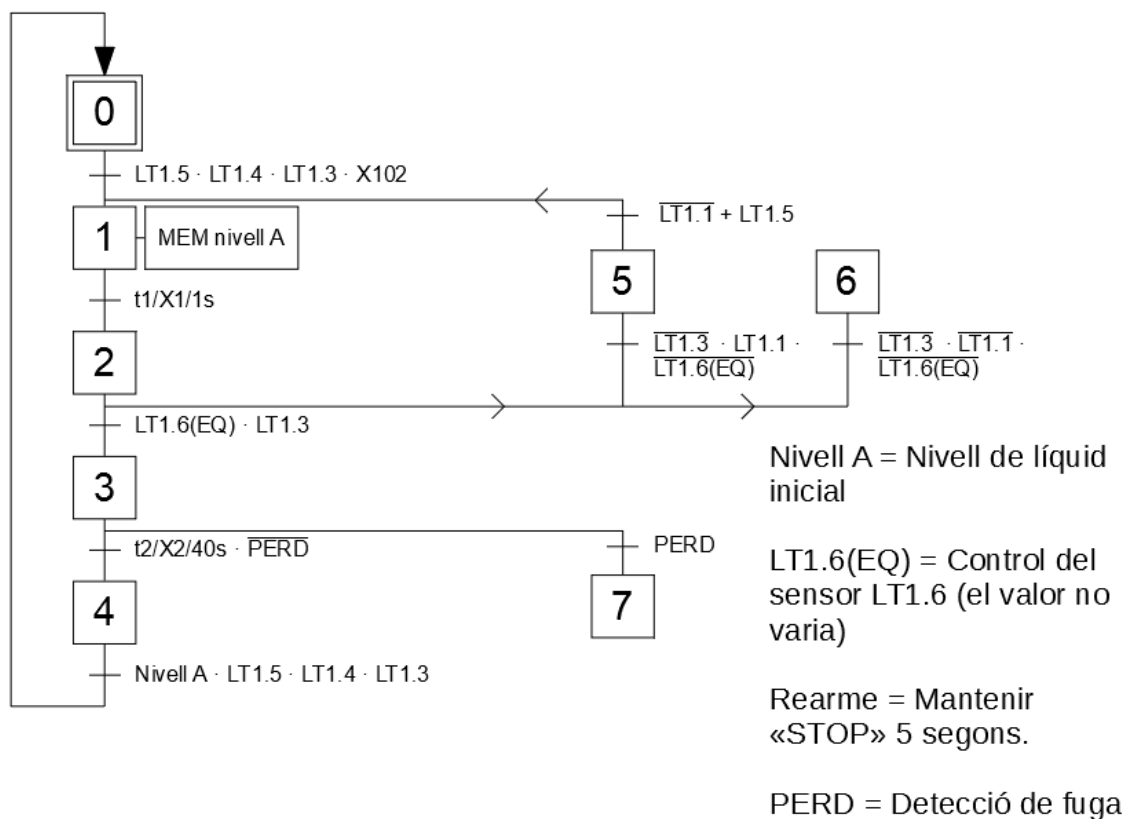
Un cop plantejat el Grafcet cal considerar la botonera que actuarà en el mode manual. Partim del teclat d'operacions físic on hi ha dos selectors, un de manual/automàtic i l'altre de on/off, també hi trobem els pulsadors de START i STOP, i el bolet d'aturada d'emergència. A més a més s'hi incorpora un selector per cada actuador (OV1.1, OV1.2, OV1.3, OV1.4, OV1.5 i OP1.1) que ens permetrà el control total de l'estació. El teclat d'operacions (imatge 72) posteriorment serà virtual fent servir un objecte de dades visuals.

TECLAT D'OPERACIONS



Imatge 72 - Teclat d'operacions segons la guia GEMMA del programa 5.

Perquè el sistema faci el procés descrit en el programa 4, s'agafa el mateix Grafcet aplicant-hi alguns canvis. Totes les accions que actuaven sobre un actuador s'eliminen perquè passaran a actuar en diferents modes de funcionament. També canvia la condició d'inici de cicle "START" que es canvia per l'estat de GEMMA en mode automàtic (X102), i els estats d'avaria o alarma (E5 i E6) passen a no tenir transicions cap a l'etapa 0 ja que abans s'haurà d'activar el mode d'avaria del Grafcet d'emergència. El Grafcet resultant del procés de neteja del vas es mostra en la imatge 73.



Imatge 73 - Grafcet en el mode automàtic del programa 5.

Els actuadors passen a tenir un control individual amb condicions segons els estats de les etapes en mode automàtic o segons el control en mode manual. Cada control individual està format únicament per dues etapes i dues transicions perquè només s'hi aplica un control tot o res sense sensors de límits en els actuadors.

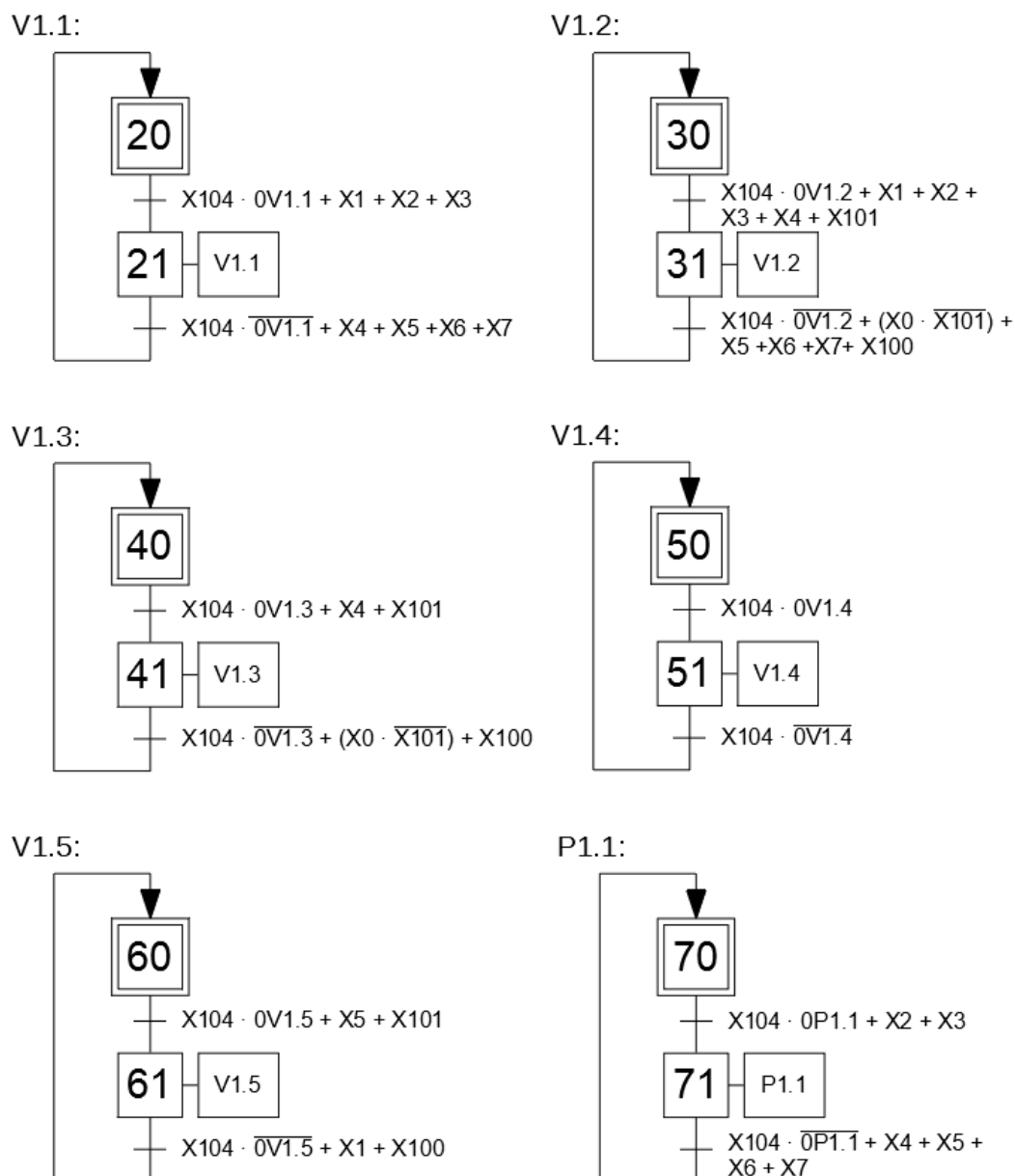
En la primera transició s'hi apliquen les condicions de control en l'estat automàtic del sistema incorporant l'estat de l'etapa on s'activa l'actuador, per exemple: si l'electrovàlvula V1.1 s'activa en l'etapa E1, en la primera transició del Grafcet de control individual s'hi incorpora la condició "X1", és a dir, quan es compleixi que el Grafcet de procés es trobi en l'etapa 1. A més a més, també s'hi incorpora la condició de quan el sistema es trobi en manual i s'activi el selector de l'actuador del teclat d'operacions (Exemple: "X104 · OV1.1"), l'actuador s'activi,.

En la segona transició hi apareixerà la condició que desactivi l'actuador, que ja pot ser per l'estat d'una etapa del mode automàtic del sistema, per exemple: si l'electrovàlvula V1.1 s'activa en les etapes E1, E2 i E3, en la segona transició s'hi incorpora la condició dels següents estats de l'etapa del mode automàtic on ja no s'activa l'actuador, en aquest cas l'actuador es pot desactivar en les etapes E4 o E5 o E6 o E7 ("X4 + X5 + X6 + X7"). També

hi apareixerà la condició del selector de l'actuador del teclat d'operacions s'ha desactivat en mode manual (Exemple: " $X104 \cdot \overline{OV1.1}$ ").

En els Grafcets dels actuadors V1.2, V1.3 i V1.4 a més a més s'incorpora a les primeres transicions o activació dels actuadors, l'estat de l'etapa "X101", referent a portar el sistema a l'estat inicial (A6), i a la desactivació o les segones transicions s'incorpora l'estat de l'etapa "X100", referent al fet que GEMMA ja acabat A6 i entra en parada a l'estat inicial (A1), i també s'incorpora en les desactivacions d'aquests actuadors l'estat E0 junt amb la no activació E101 (" $X0 \cdot \overline{X101}$ ") perquè no s'activi i es desactivi alhora.

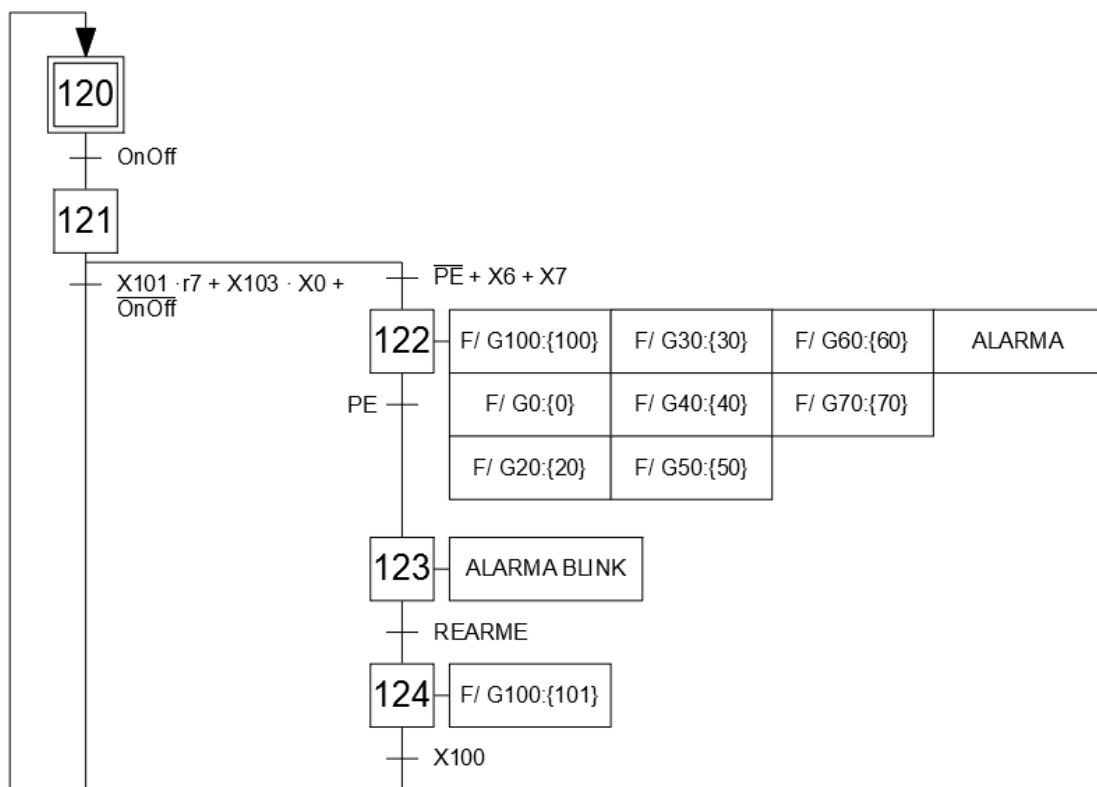
Els Grafcets de control individual dels actuadors es mostren en la imatge 74.



Imatge 74 - Grafcets individuals dels actuadors del programa 5.

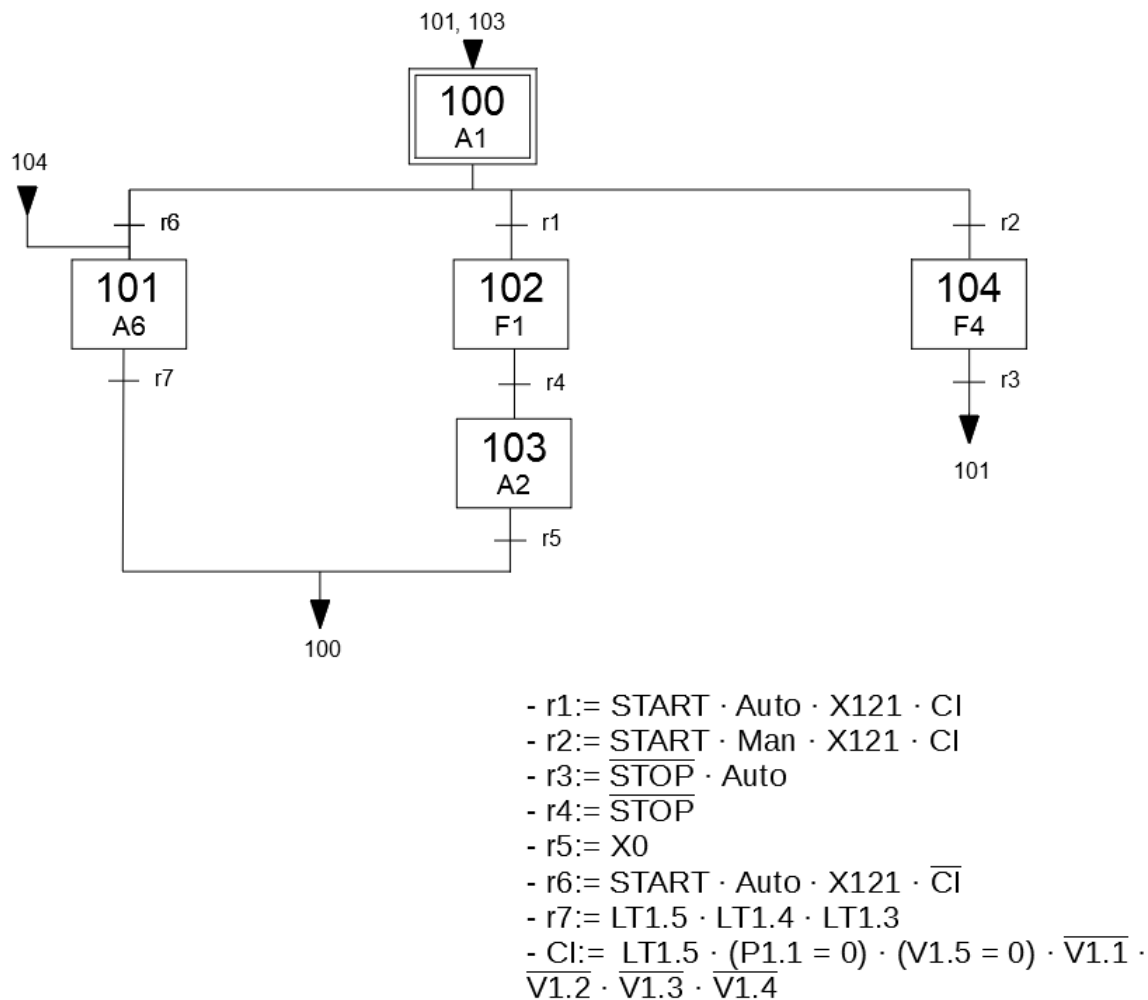
Un cop definit els Grafcets dels actuadors, es pot definir com funcionarà l'estat d'emergència. Aquest es mantindrà en l'etapa d'activació del sistema "E121" quan el selector de "on/off" es trobi en "on", a no ser que retorni el GEMMA a l'estat inicial amb les condicions de "X101 · r7" o "X103 · X0", o bé el selector d'"on/off" es trobi en "off" i es torni a posar en "on". Si l'estat d'emergència es troba en l'etapa d'activació "E121" s'encén el llum de "POWER", en cas contrari es queda apagat.

Si donem aturada d'emergència o bé entra en els estats d'alarma "E6" o "E7" entrarà en la rutina d'emergència "E122" on forçarà tots els Grafcets a les etapes inicials (E0, E100, E20...). En cas que s'ha donat aturada d'emergència i no es rearmi l'aturada d'emergència, es mantindrà el llum "ALARM" fixe, en cas que es rearmi o ha entrat a la rutina d'emergència degut a les etapes "E6" o "E7", el llum "ALARM" parpellejarà ("E123"). Per sortir d'aquesta etapa s'haurà d'activar "Rearme" i seguidament forçarà l'estat de posta en l'estat inicial ("E101") des de l'etapa "E123". Un cop en l'estat inicial "E100" sortirà de la rutina d'emergència i continuarà el funcionament normal. El Grafcet d'emergència es mostra en la imatge 75.



Imatge 75 - Grafcet d'aturada d'emergència del programa 5.

Per finalitzar el programa només falta definir les transicions del GEMMA (imatge 76):



Imatge 76 - Grafcet de la guia GEMMA amb les transicions definides del programa 5.

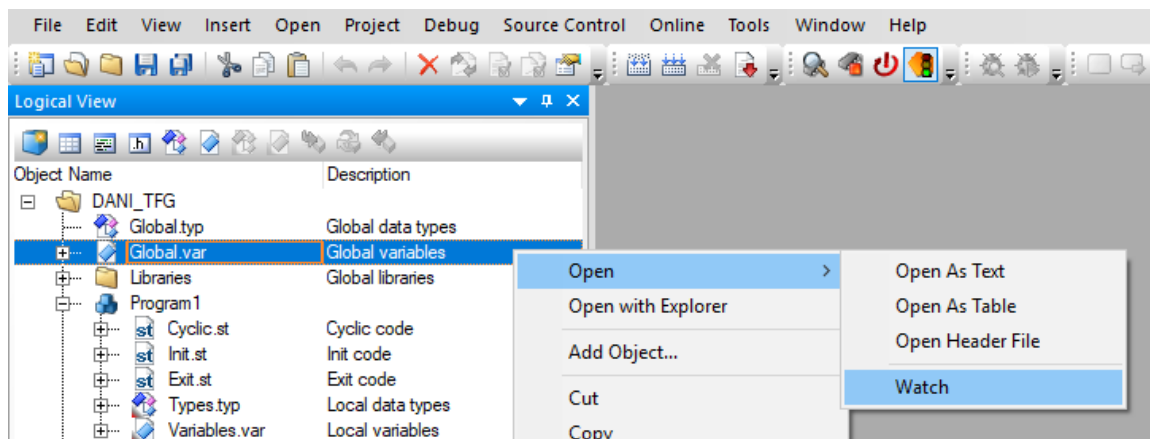
6.3. Regulació de la pressió

La regulació de pressió es farà mitjançant una lectura contínua del sensor de pressió PT1.1 i variant el valor de la bomba per obtenir una consigna desitjada en el sensor de pressió. Per obtenir una acció correctora enfront d'un canvi de consigna del sensor de pressió cal dissenyar un controlador PID que proporcioni estabilitat al sistema estudiant el comportament transitori, l'error en règim permanent i l'estabilitat d'un canvi en la consigna de l'actuador, en aquest cas la bomba, i aplicar les accions proporcional, integral i derivativa per corregir l'error en la consigna (apartat 1.1.3).

Per fer la regulació de pressió s'implementarà un controlador PID basat en un estudi anterior realitzat amb el mateix actuador (P1.1) i el mateix sensor (PT1.1) (Servitja, 2018).

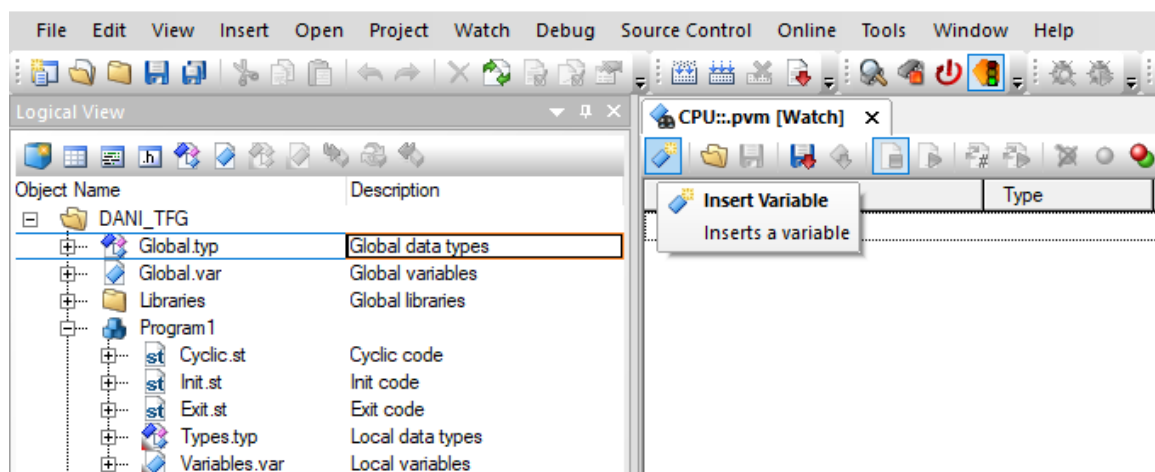
6.3.1. Monitorització de variables

Una eina de visualització d'Automation Studio és l'eina "Watch" (imatge 77), una eina que ens permet observar i/o forçar les sortides digitals i analògiques que permetrà modificar les variables I/O del programa per posteriorment adquirir les dades. Per accedir a l'eina "Watch", seleccionem l'arxiu que volem monitoritzar, en aquest cas com les entrades i sortides tant digitals com analògiques estan definides com a variables globals anem a l'arxiu nomenar per defecte com "Global.var" fem clic dret a l'arxiu, clic esquerre a open i seleccionem "Watch" (imatge 77).



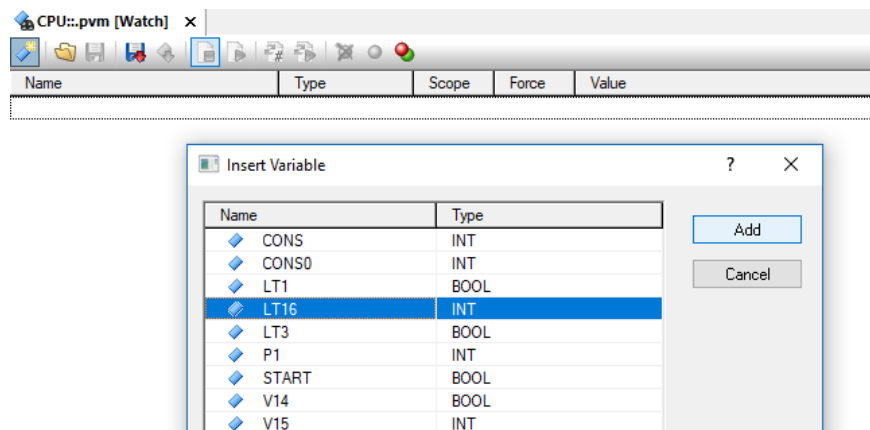
Imatge 77 - Ubicació de l'eina "Watch".

Incorporem una nova variable amb l'opció que ens apareix a dalt a l'esquerra de la finestra (imatge 78).



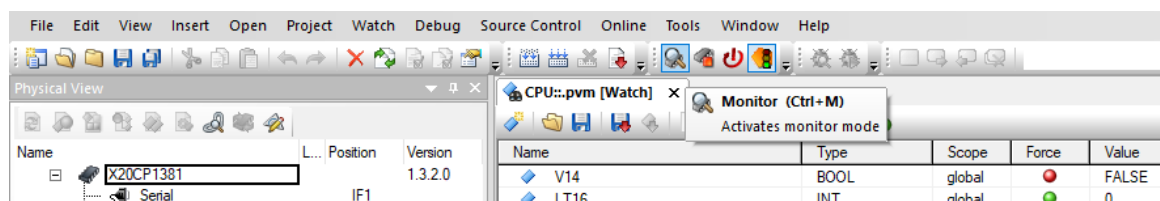
Imatge 78 - Insertar variables en l'eina "Watch".

Seleccionem quina variable volem i l'agreguem amb el botó "Add" (imatge 79):



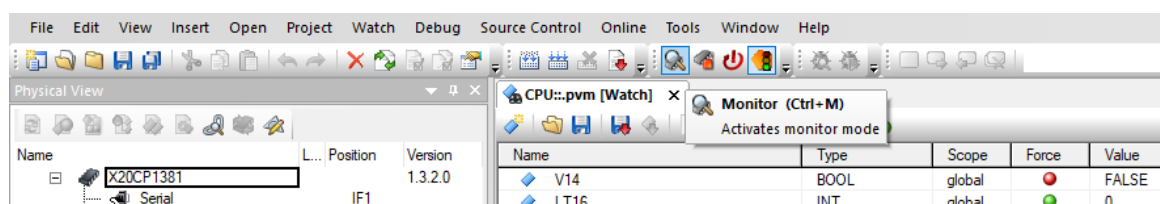
Imatge 79 - Agregar variables a l'eina "Watch".

Aquestes variables les visualitzarem quan hi carreguem un programa o cliquem a l'opció "monitor" (Imatge 80).



Imatge 80 - Activar el mode "monitor".

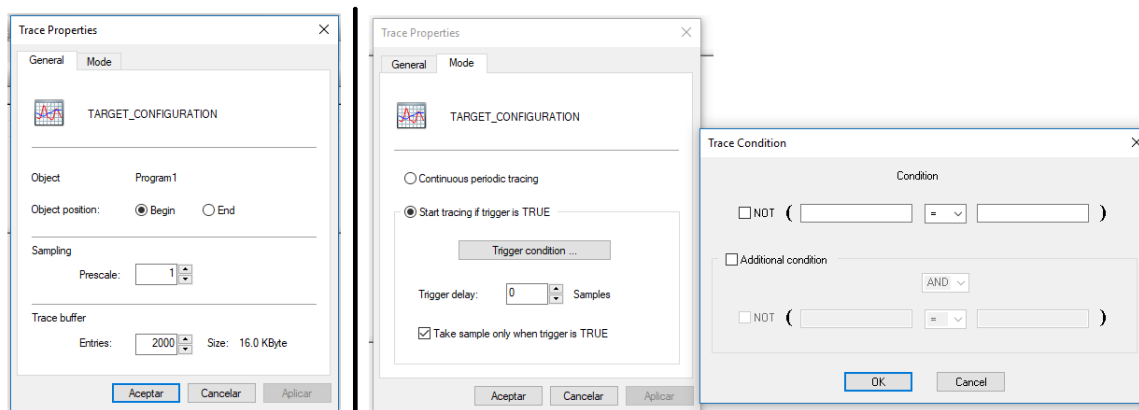
Per controlar variables de sortida o modificar els senyals d'entrada que estem visualitzant en el "Watch" cliquem sobre la boleta de la secció de "Force" i modifiquem el valor de la variable en la secció de "Value". Quan es força una variable d'entrada ho marcarà amb un cercle verd, i quan es força una variable de sortida ho marcarà amb un cercle vermell. Si es volgués retornar el control de la variable al programa, tornem a clicar sobre la icona "✖" per eliminar el forçat (imatge 81).







Imatge 81 - Forçar les variables des de l'eina "Watch".



Per visualitzar els canvis de consigna o l'evolució d'una variable es fa servir l'eina "Trace". Per accedir a aquesta eina s'accedeix a un dels programes i amb botó dret es selecciona "Open → Trace" on s'obrirà la finestra "Nom_programa [Trace]". Un cop en aquesta finestra, afeguem un "Trace" amb la icona "📄" i fent clic dret sobre la configuració del "Trace" que apareixerà amb el nom per defecte de "TARGET_CONFIGURATION" i accedim a propietats. S'obrirà la finestra propietats en l'apartat "General" on es pot configurar la posició

de l'objecte, la pre-escala de les mostres i les mostres "Entries" que es volen (una mostra per temps del "Cyclic" del programa). En l'apartat "Mode" es pot seleccionar si es vol una adquisició contínua o que adquireixi dades quan una condició és veritat amb un retard d'adquisició de mostres (imatge 82).




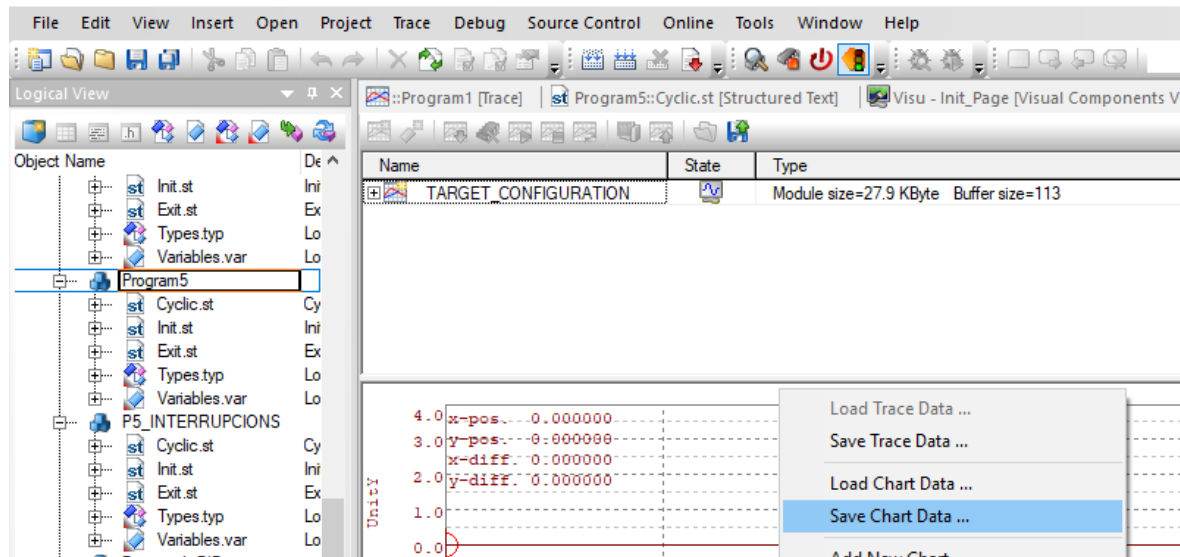
Imatge 82 - Propietats de l'eina "Trace".

Seguidament es selecciona la configuració del "Trace" que es vol i amb l'icona "Insert a new variable -  " s'agrega una de les variables del programa que es vol visualitzar. Per iniciar l'adquisició es prem l'icona "Install -  " i començarà a executar el "Trace" (sempre que el PLC estigui en funcionament amb I/O habilitades). Quan el "Trace" estigui en funcionament, es pot observar quan ocupa aquest "Trace" i el tamany del seu "Buffer" (mostres) a temps real (imatge 83). Si es desitja parar-lo es pot fer amb l'icona " ". Per desinstal·lar-lo es prem l'icona " " (imatge 83).

Name	State	Type
 TARGET_CONFIGURATION		Module size=27.9 KByte Buffer size=687

Imatge 83 - Estat del "Trace".

Per veure el gràfic resultant, un cop em parat el "Trace", es prem l'icona " " i apareixerà el gràfic de les variables inserides en la configuració del "Trace". Per guardar el gràfic fem clic dret sobre el gràfic i seleccionem l'opció "Save Chart Data..." (imatge 84) i seguidament apareix una finestra per seleccionar el directori de destí d'on desar l'arxiu de dades (extensions possibles: Char Data Files [.mch], MATLAB Binary Format [.mat], CSV [.csv], text MATLAB o Text Mathcad [.txt]).



Imatge 84 - Guardar gràfic en un arxiu de dades.

6.3.2. Programació de PID's

Per programar un controlador PID en Automation Studio es pot fer incorporant les equacions d'acció proporcional, integral i derivativa manualment o bé es pot fer servir blocs de funcions.

En cas d'incorporar manualment les equacions de les accions del PID a programa, cal tenir present la funció de cada acció:

- Acció proporcional:

$$u_P(t_k) = K_P(b \cdot y_{sp}(t_k) - y_{pv}(t_k)) \quad (1)$$

- Acció integral:

$$u_I(t_k) = u_I(t_{k-1}) + K_P \frac{T_s}{T_I} e(t_k) \quad (2)$$

- Acció derivativa:

$$u_D(t_k) = \frac{T_D}{T_D + T_s N} u_D(t_{k-1}) + \frac{K_P T_D N}{T_D + T_s N} (e_D(t_k) - e_D(t_{k-1})) \quad (3)$$

On:

$$e_D(t) = c \cdot y_{sp}(t) - y_{pv}(t), \text{ amb } c \in [0, 1] \quad (4)$$

On t_k és la mostra l'instant actual, t_{k-1} és la mostra en l'instant anterior, y_{sp} és la consigna, y_{pv} és el valor del sensor, K_P és la constant proporcional, T_I és el temps integral, T_D és el temps derivatiu, T_s és el període d'adquisició ("Cyclic" del programa) i N és el valor del filtre (normalment 10 o 100). Les constants "b" i "c" serveixen per ajuda a regular la resposta del

controlador però no afecten l'estabilitat, on els valors propers a zero generen una resposta més lenta davant dels canvis de consigna i incrementen l'estabilitat.

Per saber les constants de temps s'observa la taula de resultats obtinguda per l'anterior projecte (Servitja, 2018) fent servir el mètode empíric (taula 20).

Taula 20 - Resultats de les variables de control PID

	Termes funció de transferència en llaç obert			Paràmetres del control PID		
Voltatge Experiment	k	ξ	ω_n	K_p	T_i	T_d
5V	0,73	0,57	2,59	5,62E-01	3,50	0,9889

Per implementar el control PID en el programa, es representa un exemple en llenguatge Text Estructurat (ST) (imatge 85).

```

IF (V_PID = TRUE) THEN //Condicció per iniciar el controlador PID.

    //Actualització d'estats per modificar les variables.
    TD:=TD; //Actualització de temps derivatiu.
    TI:=TI; //Actualització de temps integral.
    KP:= KP; // Actualització de constant proporcional.

    ED:= (c * CONS_NPT11 - PT11V); //Error derivatiu.
    ERR:= CONS_NPT11 - PT11V; //Error actual (consigna - valor del sensor).

    UP:= KP * (b * CONS_NPT11 - PT11V); //Acció proporcional.
    UI:= UI_1 + (KP * (TEMPS/TI)*ERR); //Acció integral.
    UD:= ((TD/(TD + TEMPS*N))*UD) + (((KP*TD*N)/(TD+TEMPS*N))*(ED - ED_1)); //Acció derivativa.

    IF (UI_1 > 10.0) AND (UI_1 < 0.0) THEN //Limitació de l'acció integral entre 0 i 10V.
        UI := UI_1;
    ELSE
        UI:= UI;
    END_IF;

    YV:= (UP + UI + UD); //Sortida del senyal de control de tensió de la bomba.
    Y:= (UP + UI + UD) * (32767.0/10.0); //Sortida del senyal en bits.

    PT11V_1:=PT11V; //Assignació del valor del sensor en volts en l'instant anterior.
    ED_1:= ED; //Assignació de l'error derivatiu en l'instant anterior.
    UI_1:= UI; //Assignació de l'acció integral en l'instant anterior.

ELSE //Aturada del control PID quan no es compleix la condició V_PID
    Y:=0;
    ED:=0;
    ED_1:=0;
    PT11V:=0;
    PT11V_1:=0;
    UI:=0;
    UD:=0;
    UP:=0;

END_IF;

```

Imatge 85 - Exemple de programa pel controlador PID.

El controlador PID es pot definir mitjançant el bloc de funció “LCRPID ()” (on els càlculs són amb variables del tipus real o “REAL”) o bé amb la funció “LCPID ()” (on els càlculs són amb variables del tipus sencer o “INT”), junt amb la funció de configuració del PID “LCRPIDpara ()” o “LCPIDpara ()” (annex 10.3).

Per obtenir més informació accedir a “Help Explorer → Programming → Libraries → Closed loop control and mathematics → LoopConR → Function blocs and functions → LCRPID() / LCRPIDpara()” en Automation Studio.

6.3.3. Sintonia dels paràmetres del regulador PID

Per sintonitzar els paràmetres del PID, és pot utilitzar el mètode empíric, o bé utilitzant un bloc de funcions que incorpora la sintonia automàtica de paràmetres del controlador PID.

Per implementar la sintonia de paràmetres del controlador PID utilitzant el mètode empíric, s'han de fer diverses proves amb les diferents tensions aplicades a la bomba. Aquesta informació s'extraurà de l'anterior projecte (Servitja, 2018) (taula 23).

Taula 21 - Resultats de la sintonia de paràmetres de control PID

	Termes funció de transferència en llaç obert			Paràmetres del control PID		
Voltatge Experiment	k	Xi	wn	Kp	Ti	Td
4V	0,58	0,41	2,63	5,85E-01	2,08	1,8844
5V	0,73	0,57	2,59	5,62E-01	3,50	0,9889
6V	0,88	0,61	2,80	4,10E-01	1,29	1,7720
7V	1,05	0,66	2,50	2,13E-01	3,75	1,0460
8V	1,24	0,75	2,39	2,82E-01	2,45	1,6170

Aquets paràmetres s'han d'incorporar en el programa on s'implementi el controlador PID. Una manera d'incorporar la sintonia de paràmetres manual al programa és fent servir una sentència "CASE" on segons el voltatge de sortida del sistema, es seleccioni diferents valors als paràmetres del controlador PID.

Per sintonitzar els paràmetres del PID fent servir el bloc de funció "LCRPIDTune ()" (on els càlculs són amb variables del tipus real o "REAL") o bé amb la funció "LCPIDTune ()" (on els càlculs són amb variables del tipus sencer o "INT"). Els paràmetres de control adequats per al bloc de funció es calculen excitant el sistema a la vibració o iniciant un canvi del valor de la variable del sistema i registrant la variable del control. Si cal, també es pot corregir manualment (annex 10.3).

Per obtenir més informació accedir a "Help Explorer → Programming → Libraries → Closed loop control and mathematics → LoopConR → Function blocs and funcions → LCRPID() / LCRPIDpara()" en Automation Studio.

6.3.4. Automatització completa del sistema de pressió

Per automatitzar el sistema de pressió es crea dos programes nous de control de pressió.

➤ **Programa 1 PID: Control de pressió**

Pel control de pressió de l'estació, es parteix del dipòsit inferior ple com a condició inicial. El dipòsit inferior es buidarà junt amb el vas d'expansió fins que aquests arribin a un nivell mínim en el vas d'expansió i posteriorment s'omple el vas d'expansió amb una quantitat d'aigua desitjada. Seguidament s'inicia el control de pressió dins del vas d'expansió bombant aigua cap al vas d'expansió durant un cert temps. Finalment si es prem aturada, retornem el nivell del dipòsit inferior al nivell màxim, sinó es retornarà al nivell d'aigua mínim en el vas d'expansió i es tornarà a fer el control de pressió.

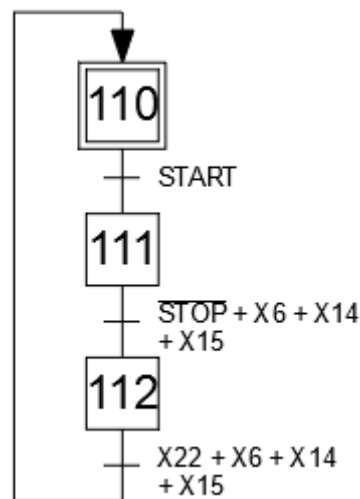
El programa consistirà en una rutina d'inici, una rutina de funcionament normal i una rutina de posta a l'estat inicial, tot això controlat per una rutina de control mitjançant els pulsadors de "START" i "STOP".

El Grafcet que executa la rutina de control controla l'estat de marxa o aturada. Aquest Grafcet compta amb 3 etapes (imatge 86):

E110- En l'etapa 110 o etapa de repòs, on no hi ha cap acció, evoluciona a la següent etapa quan es prem sobre el pulsador de "START".

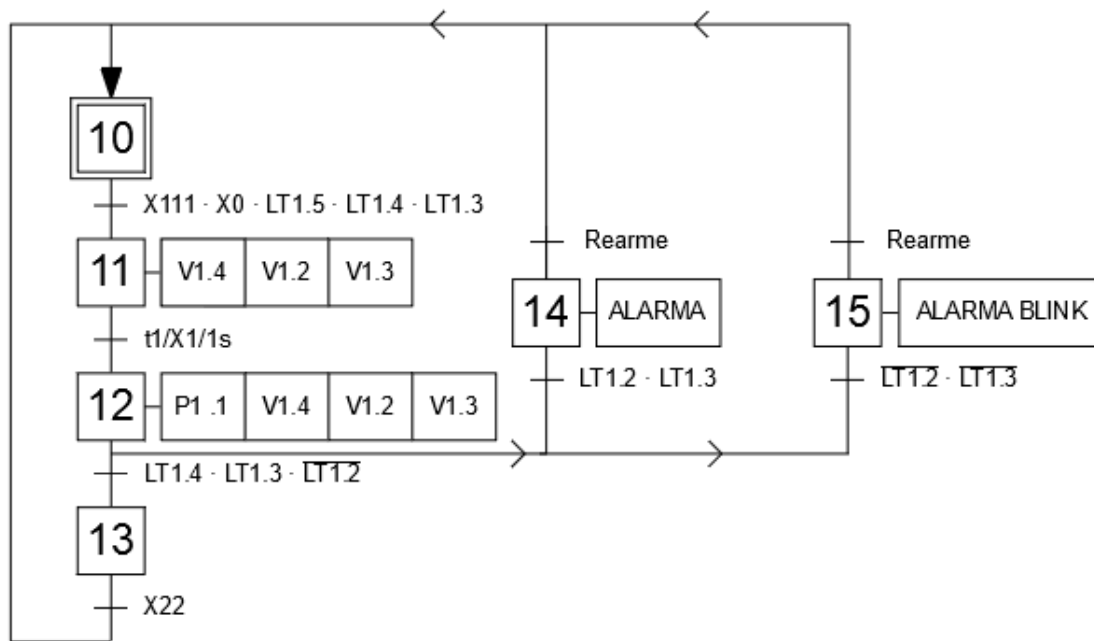
E111- En l'etapa 111 o etapa de marxa, indica quan el programa es troba en mode de funcionament. Evolucionarà d'etapa quan es prem sobre el pulsador "STOP" o es troba en alguna etapa d'alarma (X6, X14 o X15).

E112- En l'etapa 112 o etapa d'aturada es mantindrà a no ser que el programa es trobi al final de l'etapa de buidat o en alguna etapa d'alarma (X6, X14 o X15).



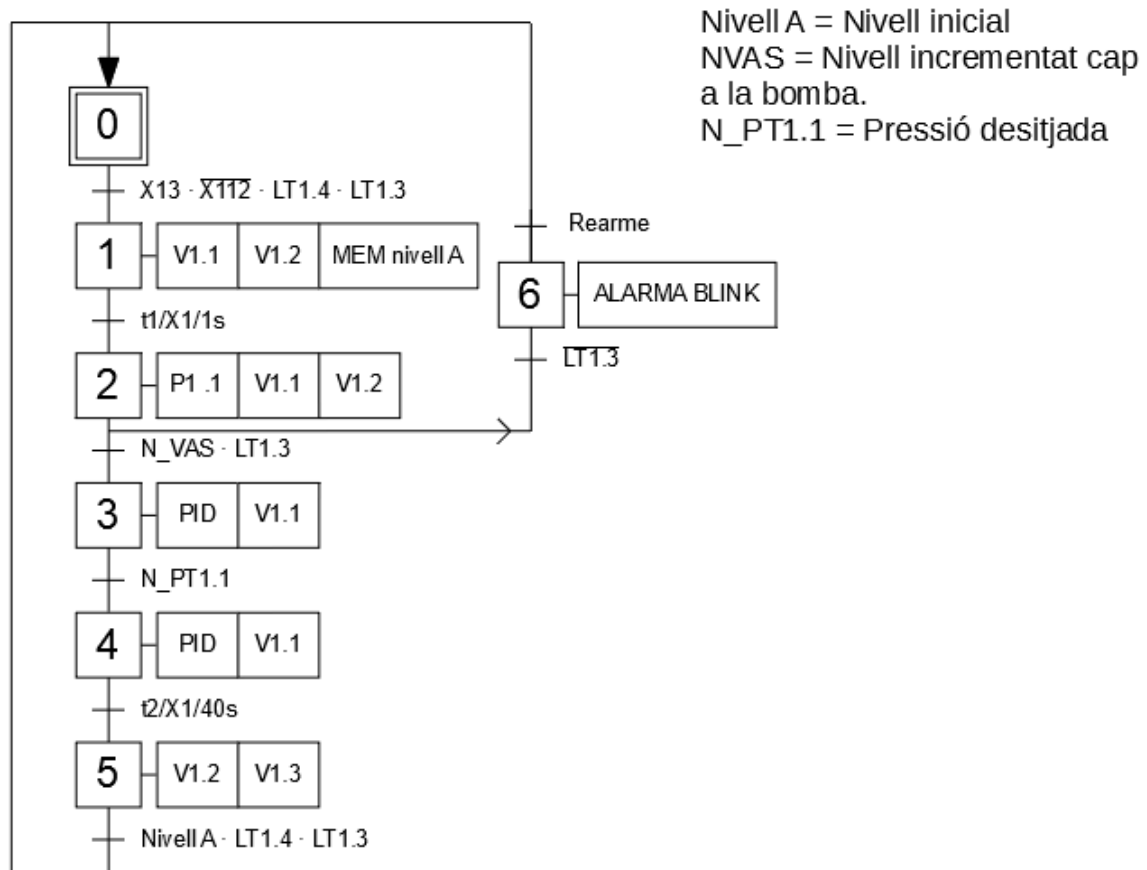
Imatge 86 - Grafcet de rutina de control del programa1_PID.

La rutina d'inici té com a objectiu preparar l'estació pel control de pressió, es a dir deixar el dipòsit al nivell mínim del vas (LT1.4). Per iniciar el programa, la rutina de control es te de trobar en marxa "E111", la rutina de funcionament normal en repòs "E0" i detectant els 3 sensors de nivell (LT1.5, LT1.4 i LT1.3). Durant el transcurs d'aquesta rutina es bombarà aigua del dipòsit inferior cap al dipòsit superior mitjançant la bomba fins que el nivell d'aigua del dipòsit inferior estigui al nivell del sensor LT1.4. Durant el transcurs del bombeig d'aigua s'incorporen dos sistemes d'alarma de límits de nivell, una alarma per quan el nivell del dipòsit superior es trobi ple (LT1.2* → "E15") i un per quan el dipòsit inferior estigui buit (LT1.3* → "E14"). Quan la rutina d'inici es completi, aquesta es quedarà en un estat d'espera "E13" d'on sortirà quan la rutina de posta a l'estat inicial hagi acabat ("E22"). En la imatge 87 es mostra el Grafcet de funcionament.



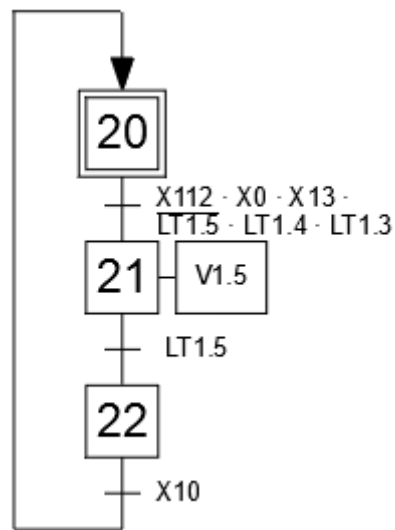
Imatge 87 - Grafcet de rutina d'inici del programa1_PID.

La rutina de funcionament normal (imatge 88) s'iniciarà amb un nivell inicial d'aigua "Nivell_A" i quan la rutina d'inici es trobi en l'estat d'espera "E13", la rutina de control no es trobi en l'estat d'aturada "E112" i es detectin els sensors capacitius LT1.4 i LT1.3. En el programa es podrà definir un nivell d'aigua "N_VAS" que es bombejarà cap al vas d'expansió. En el transcurs d'aquest bombeig d'aigua s'incorpora un sistema de detecció d'alarma en cas que el dipòsit inferior estigui buit (LT1.3* → "E6"). Un cop el vas d'expansió ja s'hagi omplert amb una quantitat d'aigua definida per "N_VAS" es procedeix al control del sensor de pressió mitjançant el controlador PID. Primer s'arribarà a la consigna de pressió que es vulgui "N_PT1.1" i es mantindrà durant un temps "t2" de 40 segons on es podrà variar la consigna "N_PT1.1" manualment amb l'ajuda d'un visualitzador creat amb Automation Studio. Quan el temporitzador finalitzi, es tornarà a deixar el nivell inicial que hi havia en la rutina de funcionament normal. La rutina de funcionament normal seguirà funcionant fins que la rutina de control es trobi en estat d'aturada "E112*".



Imatge 88 - Grafcet de rutina de funcionament normal del programa1_PID.

La rutina de posta a l'estat inicial s'executarà quan la rutina de control es trobi en l'estat d'aturada "E112", la rutina de funcionament normal es trobi en estat de repòs "E0", la rutina d'inici es trobi en estat d'espera "E13" i es detecti LT1.4 i LT1.5. Un cop s'iniciï la rutina, s'obrirà l'electrovàlvula V1.5 fins que el dipòsit inferior estigui ple (LT1.5) i entrarà en l'etapa de finalització de rutina de posta en l'estat inicial "E22". Quan detecti que la rutina d'inici es troba en repòs ("E10"), aquest també retornarà a l'estat de repòs ("E20") (imatge 89).



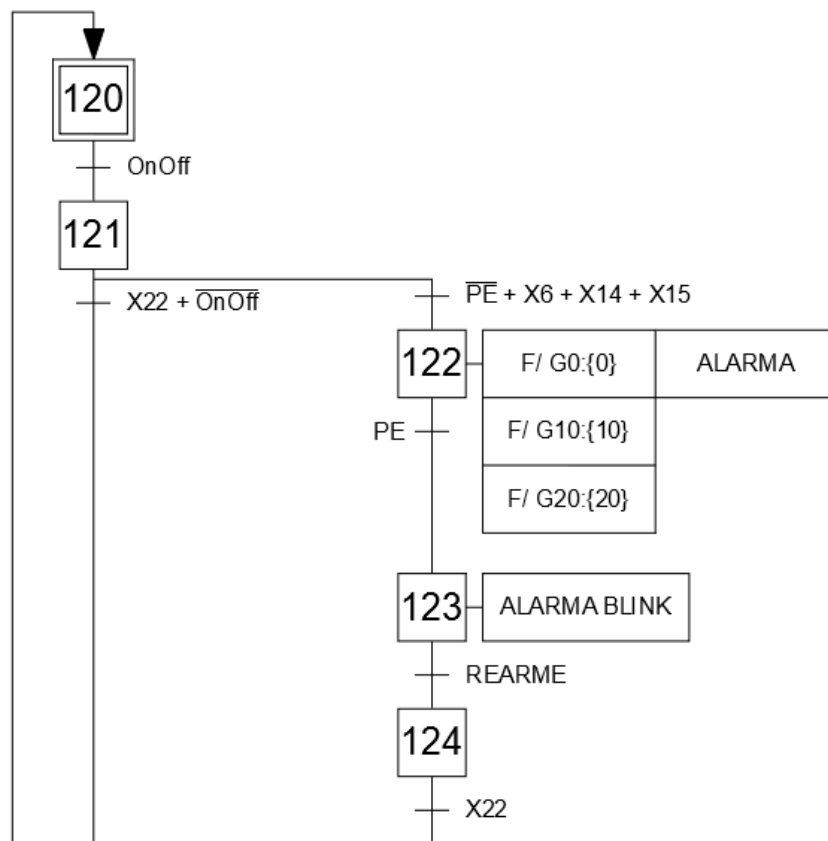
Imatge 89 - Grafcet de rutina de posta en l'estat inicial del programa1_PID.

Si alguna rutina entrés en un estat d'alarma, es fa un "REARME" (mantenir 5 segons el pulsador de stop) perquè aquestes retornin a l'estat de repòs ("E0" o "E10") de la rutina a la qual correspon.

➤ **Programa 2 PID: Control de pressió amb aturada d'emergència**

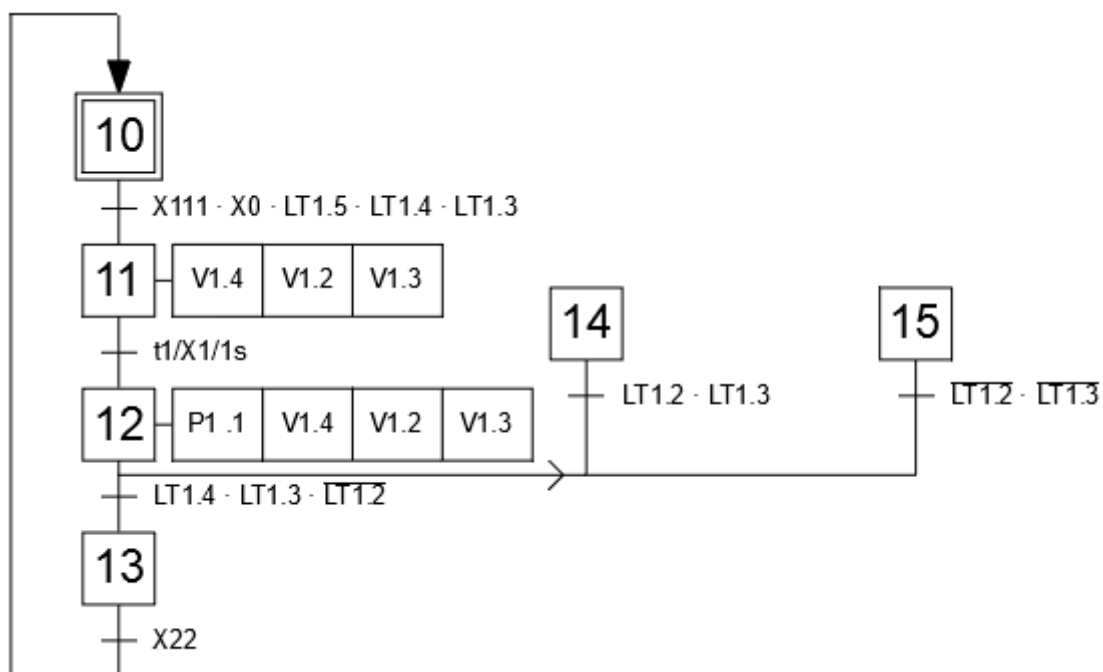
En aquest programa s'incorpora en el control de pressió una rutina d'emergència (sense aplicar un diagrama GEMMA) i es modifiquen les rutines de l'anterior programa per sincronitzar-se amb la rutina d'emergència.

La rutina d'emergència (imatge 90) inicialment es troba en repòs "E120" fins que el selector "On / Off" es trobi en la posició "On" on llavors passarà a l'estat de funcionament "E121" (tornarà a l'estat de repòs si el selector està en "Off" o si la rutina de posta en l'estat inicial es troba en l'estat de finalització "E22"). En el cas que el bolet d'emergència s'ha activat o una de les rutines es troba en estat d'alarma (E6, E14 o E15) la rutina d'emergència es posarà en estat d'emergència on es força a les rutines a l'estat de repòs ("E0, E10, E20") i s'encen un llum d'alarma si el bolet d'emergència no està rearmat o bé l'alarma fa pampallugues si el bolet d'emergència està rearmat o si s'ha entrat en un estat d'alarma. Per sortir d'aquest estat, es fa un "REARME" (mantenir 5 segons el pulsador de stop) i es posarà en estat de posta a l'estat inicial accedint a la rutina amb aquest nom. La rutina d'emergència tornarà a l'estat de repòs quan la rutina de posta a l'estat inicial es trobi en l'estat de finalització "E22".

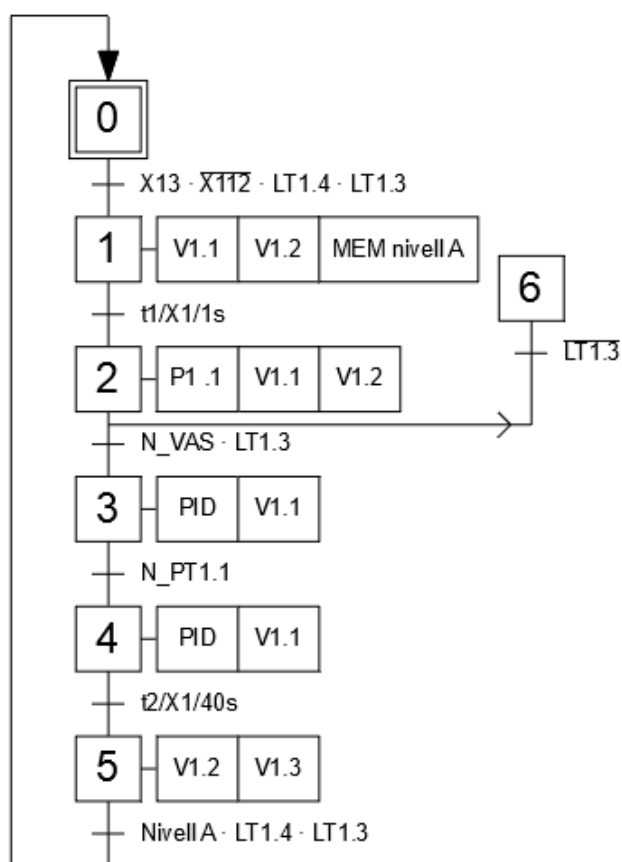


Imatge 90 - Grafcet de rutina d'emergència del programa2_PID.

Les rutines d'alarma "E6", "E14" i "E15" es quedaran en "pous" (etapes sense transicions pròpies) perquè es força els estats de repòs quan s'entra en un estat d'alarma (degut a la rutina d'emergència). Els Grafcets de les rutines d'inici i funcionament normal quedaran de la següent manera (imatge 91 i 92).



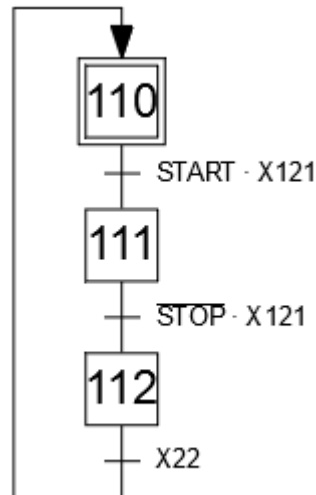
Imatge 91 - Grafcet de rutina d'inici del programa2_PID.



Nivell A = Nivell inicial
 NVAS = Nivell incrementat cap a la bomba.
 N_PT1.1 = Pressió desitjada

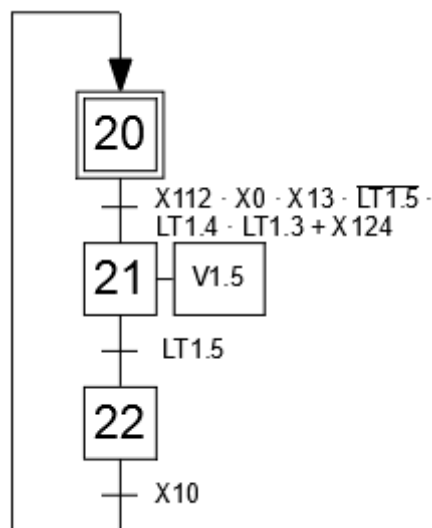
Imatge 92 - Grafcet de rutina de funcionament normal del programa2_PID.

En el Grafcet de la rutina de control es modifica l'estat de repòs "E110" i l'estat de "marxa" agregant la condició que la rutina d'emergència estigui en l'estat de funcionament "E121" (imatge 93).



Imatge 93 - Grafcet de rutina de control del programa2_PID.

En el Grafcet de la rutina de posta a l'estat inicial, s'incorpora en la transició de l'estat de repòs "E20", que en cas que la rutina d'emergència es trobi en l'estat de posta a l'estat inicial "E124" també s'executi la rutina de posta a l'estat inicial (imatge 94).



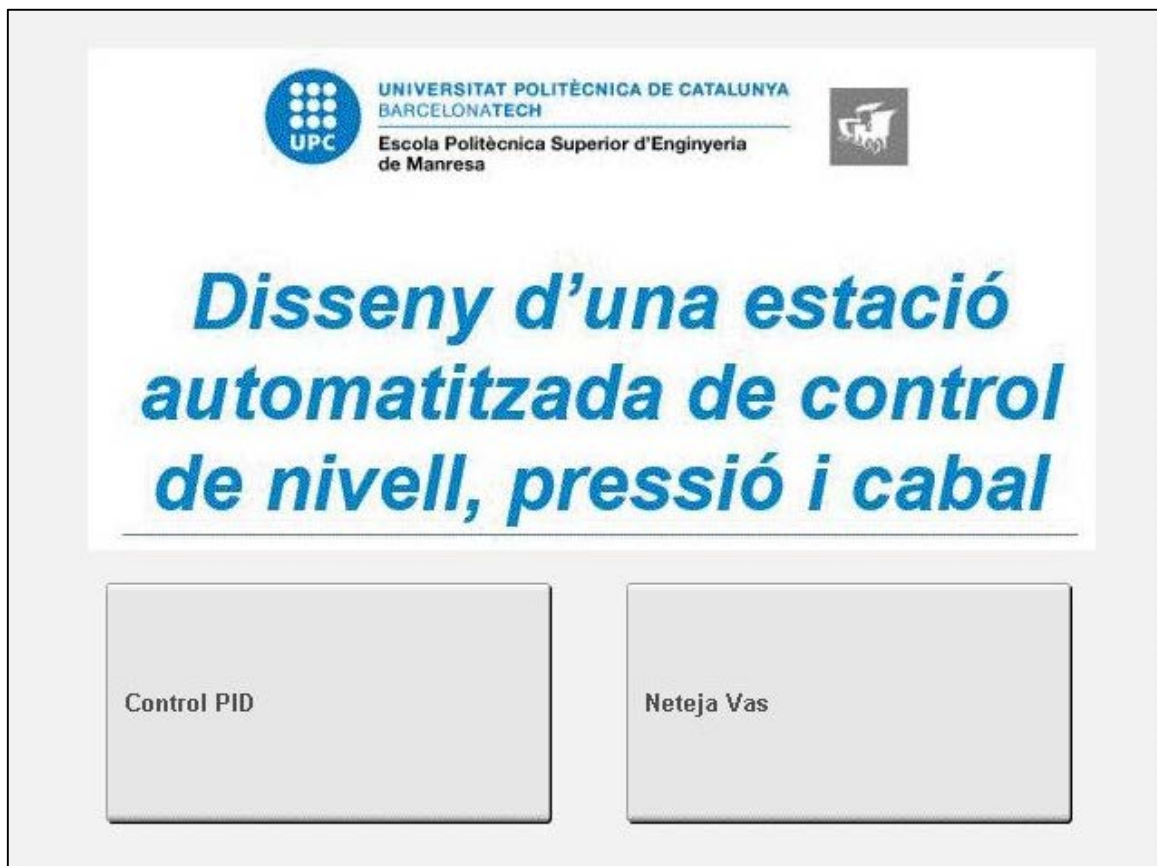
Imatge 94 - Grafcet de rutina de posta en l'estat inicial del programa2_PID.

Posteriorment s'hi pot col·locar un objecte visualitzador per controlar les variables de consigna de nivell "N_VAS" i la consigna del sensor de pressió "N_PT1.1", així com modificar el temps del temporitzador on s'executa el controlador PID ("t2").

6.4. Sistema SCADA

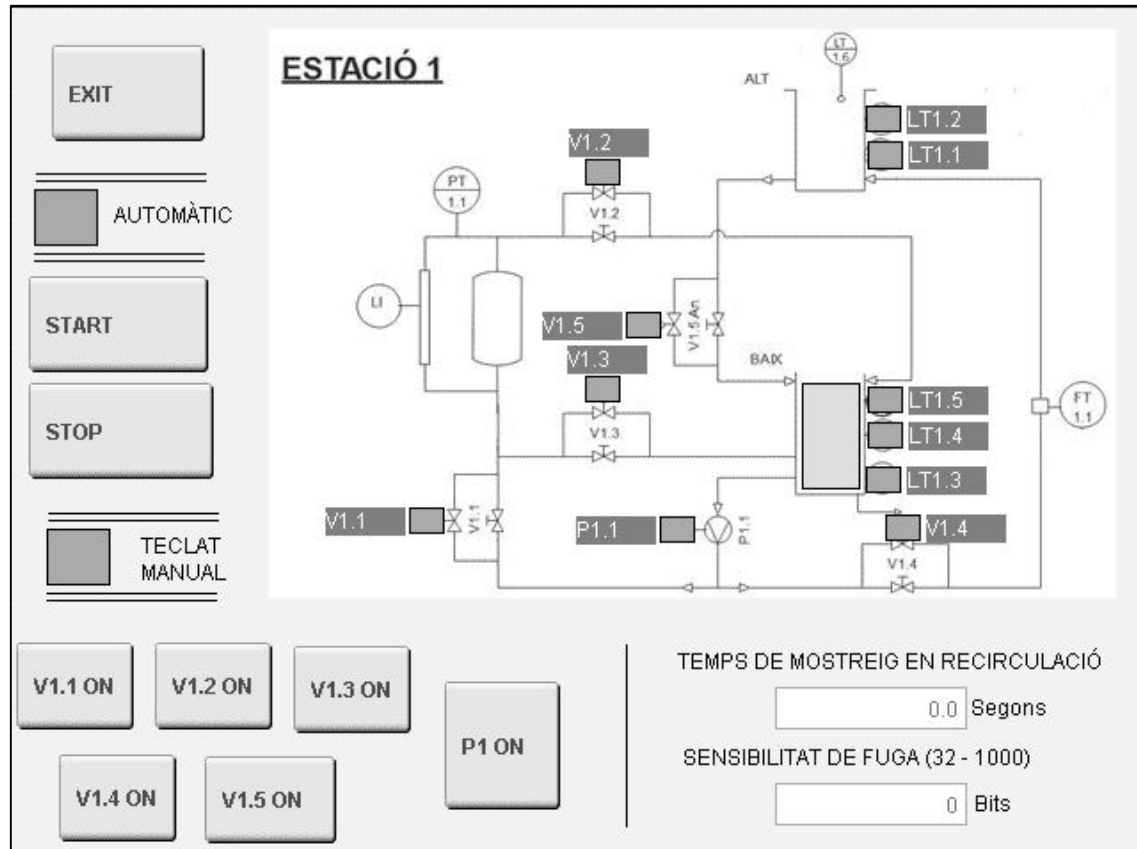
El sistema SCADA que es planteja en aquest projecte s'ha dissenyat per utilitzar-se en paral·lel amb el teclat d'operacions físic de l'estació.

Els programes que incorporen un sistema SCADA són el programa 5 (apartat 6.2.1) i el programa 2 PID (apartat 6.3.4), on s'incorpora un objecte visualitzador compartit amb els dos programes (apartat 5.6). El sistema SCADA partirà d'una finestra inicial on es pot triar el programa que executarem, el programa que es vol executar caldrà carregar-ho individualment: Control PID (programa 2 PID) o Neteja Vas (programa 5) (imatge 95).



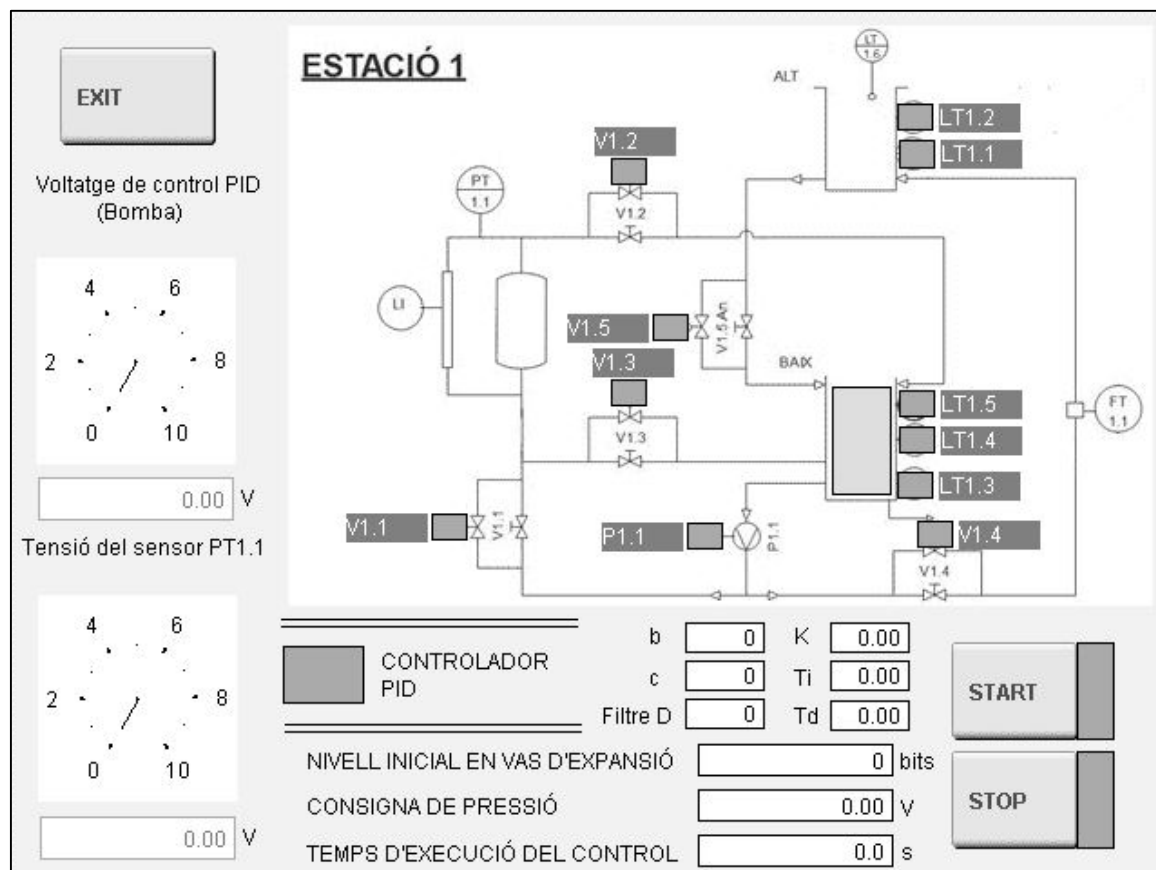
Imatge 95 - Sistema SCADA: Pantalla inicial.

Pel control del programa 5 (càrrega dels programes "Program5" i "P5_INTERRUPCIONES" en el PLC), es selecciona l'opció "Neteja Vas". Per dissenyar el teclat d'operacions virtual s'agafa com a base el teclat d'operacions plantejat en la imatge 72 i s'incorporen altres controls com ara el temps de mostreig en recirculació que incorpora el programa, i el control de la sensibilitat de fuga del programa. El visualitzador incorpora un esquema hidràulic de l'estació des d'on es poden monitoritzar els elements de l'estació 1. També es pot observar el mode de funcionament actual del programa. El teclat d'operacions resultant es mostra en la imatge 96.



Imatge 96 - Sistema SCADA: Neteja de vas (programa 5).

Pel control del programa 2 PID (càrrega dels programes “Program2_PID” i “PID_INTERRUPTIONS” en el PLC), es selecciona l’opció “Control PID”. Aquest teclat d’operacions implementa un visualitzador de la tensió de control de la bomba que s’envia al variador i un visualitzador de la tensió que indica el sensor de pressió. A més a més, es pot controlar les constants del controlador PID que s’indicarà l’activació del control amb un llum de “CONTROLADOR PID”. Les altres constants a controlar d’aquest programa són: el nivell inicial en vas d’expansió, consigna de pressió desitjada i temps d’execució del control de pressió. El visualitzador també incorpora un esquema hidràulic de l’estació des d’on es poden monitoritzar els elements de l’estació 1 i un llum en els pulsadors de “START” i “STOP” que permeten visualitzar quin dels dos pulsadors s’ha actuat. El teclat d’operacions resultant es mostra en la imatge 97.



Imatge 97 - Sistema SCADA: Control PID (programa 2 PID).

7. PROBLEMES I SOLUCIONS

7.1. Problemes de programa

En el transcurs del projecte han sorgit diversos problemes alhora de programar. Els errors més comuns causats en la programació d'Automation Studio eren deguts a la definició del tipus de variables, ja que apareixen tipus de variables assignats a blocs de funcions com ara el temporitzador "TON".

La majoria de problemes de programació han estat resolts utilitzant correctament l'ajuda que ofereix Automation Studio. El buscador "Help Explorer" ha permès acotar la cerca sobre qualsevol problema que ha sorgit durant la programació o dubtes en les característiques de funcionament d'entrades i/o sortides.

7.1. Problemes de disseny en l'estació

El principal problema a destacar és la ubicació d'algunes de les canonades en la maqueta, principalment la canonada de sortida del vas d'expansió cap al dipòsit inferior. Aquesta canonada inicialment on hi va col·locada l'electrovàlvula V1.2, es trobava en forma de "U" i això feia que quan s'acabava de recircular aigua del vas (sortida superior) cap al dipòsit inferior s'estancava l'aigua en la base de la "U" de la canonada impedit l'entrada d'aire per la part superior del vas d'expansió i fent impossible el buidatge per gravetat del vas d'expansió. Aquest problema ha causat confusions alhora de diagnosticar si l'electrovàlvula es trobava oberta o tancada o si estava espatllada la vàlvula manual col·locada en paral·lel.

La solució a aquest problema ha estat aixecar la canonada fins al nivell de la sortida superior del vas d'expansió i d'aquesta manera evitar estancs d'aigua en la canonada per on s'entra l'aire per buidar el vas d'expansió.

8. CONCLUSIONS

Aquest projecte ha ajudat a endinsar-se d'una manera tècnica en el món de l'automatització i l'indústria moderna. El fet de crear una maqueta nova i construir-la mitjançant diferents elements electrònics permet que s'adquireixin habilitats i coneixements pràctics sobre circuits elèctrics.

En la tasca de crear una estació que incorpora un control de nivell, pressió i cabal s'ha millorat l'autonomia i s'ha reduït l'espai envers l'antiga estació que incorporava aquests controls en una maqueta més gran.

Amb els diferents apartats que apareixen en el projecte, es pot familiaritzar fàcilment amb l'estació i entendre els controls de nivell, de pressió o de cabal que es poden dur a terme mitjançant el software de programació de B&R Automation Studio.

Per familiaritzar-se amb l'entorn d'Automation Studio aquest projecte presenta una guia de com funciona aquest entorn de programació i es poden veure les característiques bàsiques de les que disposa aquest software.

En cas de voler aprendre a programar amb text estructurat, aquest treball podria servir com a guia, ja que apareixen diversos programes amb els quals controlar l'estació d'una manera més simple recreant els programes per automatitzar l'estació de treball, o d'una manera més complexa recreant o utilitzant com a referència el programa 5, on s'implementa la guia GEMMA per definir diferents modes de funcionament de l'estació i els diferents estats de marxa i aturada de l'estació.

També permet conèixer i incorporar un controlador PID en qualsevol control o conjunt de controls que es vulgui aplicar a la maqueta.

Aquest projecte pot ser utilitzat en un futur com a eina didàctica per adaptar-se a sistemes automatitzats utilitzant un software i un controlador lògic programable modern.

8.1. Tasques proposades

Una idea per a futures tasques és que amb els elements restants de l'antiga maqueta (apartat 2.1), s'elabori una còpia d'aquesta maqueta i que se'n creïn de noves com per exemple una estació de control de temperatura.

Fent servir les estacions proposades junt amb una nova estació de distribució incorporant el PLC de B&R X20CP1583, es pot crear una maqueta molt més gran i amb altres controls.

Una altra tasca que es pot realitzar és la unió dels programes en un de sol amb l'ajudada de la guia GEMMA. En cas de tenir altres estacions, unir-les amb l'estació creada en aquest treball i crear un programa més extens, on es sincronitzin totes les estacions.

9. BIBLIOGRAFÍA

- Barraguer, X. M. (Octubre / 2016). *Sistema de Control Distribuido*. Consultat el Abril / 2018, a Industria Química es un portal de Infoedita: <https://www.industriaquimica.es/articulos/20161114/sistema-control-distribuido-nuevas-soluciones-adaptarse-exigencias-mercado#.Wv3BSFyWSUK>
- Cervantes, O. (sense data). *Historia PLC*. Consultat el Maig / 2018, a Academia: https://www.academia.edu/29678730/Historia_plc
- H. Bischoff, D. H. (Octubre de 1997). *Process Control System Workbook*. (F. Ebel, Ed.) Dresden, Alemania: Festo Didactic GmbH & Co. Recuperado el Març de 2018
- Javier Román. (sense data). *Automatización de Plantas Industriales: Sistemas de Control Distribuido y Controladores Lógicos Programables*. Consultat el Abril / 2018, a new.abb.com: [http://www02.abb.com/db/db0020/db002013.nsf/0/0c6eee0874b55c5cc12576f70067c380/\\$file/Art%C3%ADculo+t%C3%A9cnico+-+Electromagazine+No+34.pdf](http://www02.abb.com/db/db0020/db002013.nsf/0/0c6eee0874b55c5cc12576f70067c380/$file/Art%C3%ADculo+t%C3%A9cnico+-+Electromagazine+No+34.pdf)
- José Daniel Serrano, A. (2017). *Diseño, implementación e integración de un sistema de medición de variables ambientales en un sistema IoT con software y hardware libre*. Consultat el Maig / 2018, a Universitat Politècnica de València: RiuNet repositori UPV: https://riunet.upv.es/bitstream/handle/10251/89420/48645867Q_TFG_15047219624054883031430478186001.pdf
- Martín, F. M. (Octubre de 2016). *Autómatas Programables: Introducción al Estándar IEC-61131*. Obtenido de <http://isa.uniovi.es/>, Universidad de Oviedo: Área de Ingeniería de Sistemas y Automática): <http://isa.uniovi.es/docencia/IngdeAutom/transparencias/Pres%20IEC%2061131.pdf>
- Mazaeda, E. d. (sense data). *Industria 4.0*. Consultat el Abril / 2018, a Universidad de Valladolid: http://uvadoc.uva.es/bitstream/10324/17506/1/PID_15_156_Anexo6.pdf
- MOLLO, J. J. (2014). *INTEGRACIÓN DE DIFERENTES SISTEMAS DE CONTROL DISTRIBUIDO (DCS) UTILIZANDO SOFTWARE ESTÁNDAR Y TECNOLOGÍA OPC*. Consultat el Maig / 2018, a repositorio.unsa.edu.pe (UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA): <http://repositorio.unsa.edu.pe/bitstream/handle/UNSA/3313/IEmomojj.pdf?sequence=1&isAllowed=y>

- Navarro., D. E. (Gener de 2018). *Industria 4.0: Retos y Oportunidades en las Factorías de Automoción*. (D. F. Fluixá., Ed.) Recuperado el Maig de 2018, de Universitat Politecnica de Valencia: RiuNet repositorio UPV: <http://uvadoc.uva.es/bitstream/10324/29368/1/TFM-I-803.pdf>
- Oliva Navarro, E. D. (Gener / 2018). *Industria 4.0: Retos y Oportunidades en las Factorías de Automoción*. Consultat el Maig / 2018, a Universidad de Valladolid Biblioteca Universitaria: <http://uvadoc.uva.es/handle/10324/29368>
- Pineda Sánchez, M. (sense data). *Nuevas tendencias en la automatización con autómatas programables basados en la Norma IEC-61131-3*. Consultat el Maig / 2018, a Universitat Politecnica de Valencia: RiuNet repositorio UPV: <https://riunet.upv.es/bitstream/handle/10251/31225/Nuevas%20tendencia%20en%20la%20programaci%C3%B3n%20de%20PLCs.pdf>
- PLCopen. (s.f.). *PLC Open for efficiency in automation*. Recuperado el Mayo de 2018, de <http://www.plcopen.org>
- Reinhard Langmann, L. F.-P. (24-26 / Febrer / 2016). *A PLC as an Industry 4.0 component*. Consultat el Maig / 2018, a vrd.library.sh.cn: <http://vrd.library.sh.cn/mgr/ueditor/net/upload/2016-08-31/70def0f7-120b-4836-9a05-15fd22b93fb7.pdf>
- Servitja, J. (10 / Gener / 2018). *Desenvolupament d'eines didàctiques per a l'aprenentatge de control de processos industrials*. Manresa, Barcelona. Consultat el Maig / 2018, a <https://upcommons.upc.edu/bitstream/handle/2117/115256/Memoria%20Jaume%20Servitja.pdf?sequence=1&isAllowed=y>
- Suárez, V. M. (sense data). *Modos de marcha y parada: La guía GEMMA*. Guijón, Asturias. Consultat el Juny / 2018, a <http://isa.uniovi.es/~vsuarez/Download/GemmaTelemecanique.PDF>

10. ANNEXOS

10.1. Annex 1: Programes

10.1.1. Variables globals i Cyclic dels programes

Name ▲	Type		
ALARMA	BOOL	T123	BOOL
ALARMA_BLINK	BOOL	T124	BOOL
A_M	BOOL	TCOUNT	USINT
CONS	INT	TECLAT	INT
CONSO	INT	TON_03	TON
E0	BOOL	V11	BOOL
E6	BOOL	V12	BOOL
E7	BOOL	V13	BOOL
E14	BOOL	V14	BOOL
E15	BOOL	V15	INT
E22	BOOL		
E100	BOOL		
E101	BOOL		
E103	BOOL		
E120	BOOL		
E121	BOOL		
E122	BOOL		
E123	BOOL		
E124	BOOL		
EMERGENCIA	INT		
ESTAT	INT		
GEMMA	INT		
GRAF_FI	USINT		
GRAF_INIT	INT		
GRAF_P1	INT		
GRAF_V11	INT		
GRAF_V12	INT		
GRAF_V13	INT		
GRAF_V14	INT		
GRAF_V15	INT		
LT1	BOOL		
LT2	BOOL		
LT3	BOOL		
LT4	BOOL		
LT5	BOOL		
LT16	INT		
NALARMA	BOOL		
OnOff	BOOL		
P1	INT		
PE	BOOL		
POWER	BOOL		
PT11	INT		
r7	BOOL		
START	BOOL		
STOP	BOOL		
T120	BOOL		
T121	BOOL		
T122	BOOL		

Exception							
Cyclic #1 - [10 ms]							
P5_INTERRU	1.00.0	UserROM	3960	30/06/2...	P5_INTERRUPCIONES	Config1\X2...	
PID_INTERR	1.00.0	UserROM	0		PID_INTERRUPCIONES	Config1\X2...	
Cyclic #2 - [20 ms]							
Cyclic #3 - [50 ms]							
Program1	1.00.0	UserROM	0		Program1	Config1\X2...	
Program2	1.00.0	UserROM	0		Program2	Config1\X2...	
Program3	1.00.0	UserROM	0		Program3	Config1\X2...	
Program4	1.00.0	UserROM	0		Program4	Config1\X2...	
Program5	1.00.0	UserROM	8056	05/07/2...	Program5	Config1\X2...	
Program1_P	1.00.0	UserROM	0		Program1_PID	Config1\X2...	
Program2_P	1.00.0	UserROM	0		Program2_PID	Config1\X2...	
Cyclic #4 - [100 ms]							
Cyclic #5 - [200 ms]							
Cyclic #6 - [500 ms]							
Cyclic #7 - [1000 ms]							
Cyclic #8 - [10 ms]							

10.1.2. Programa 1

- Variables locals:

Name	Type
TON_05	TON
COUNT	BOOL
E1	BOOL
E2	BOOL
E3	BOOL
E4	BOOL
LT16A	INT
LT16A2	INT
LT16B	INT
LT16EQ	BOOL
LT16LOG	INT
N	INT
NIVELL_A	INT
PRIMER	INT
T0	BOOL
T1	BOOL
T2	BOOL
T3	BOOL
T4	BOOL
TON_01	TON
TON_02	TON
VP1	BOOL

- Programa:

```
PROGRAM _INIT
  ESTAT:=0;
  E0 := TRUE;
  E1 := FALSE;
  E2 := FALSE;
  E3 := FALSE;
  E4 := FALSE;
```

```

LT16A:=0;
LT16B:=0;
COUNT:=0;
N:=29;
PRIMER := 0;
V11:=FALSE;
V12:=FALSE;
V13:=FALSE;
V14:=FALSE;
V15:=0;
P1:=0;

END_PROGRAM

PROGRAM _CYCLIC

//CONTROL DE VARIABLES
//LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
LT16LOG := 32767 - LT16;

//CONTROL DE LT16EQ

IF (E2=TRUE) THEN
    LT16EQ := FALSE;
    COUNT := TRUE;

    IF (TON_05.Q = TRUE) THEN
        LT16A := LT16LOG;
        COUNT := FALSE;

        IF (PRIMER = 0) THEN
            LT16A2 := LT16A;
            LT16A := 0;
            PRIMER := 1;

        ELSIF (PRIMER = 1) THEN

            IF (LT16A >= LT16A2) THEN
                LT16EQ := TRUE;
                LT16A2:= LT16A;
                PRIMER := 0;
                LT16A := 0;

            ELSE
                LT16A2 := LT16A;
            END_IF;

        ELSE
            END_IF;
    ELSE
        COUNT := FALSE;
    END_IF;

//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)

IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;

// _____

// TRANSICIONS ENTRADES
T0:= START;
T1:=TON_01.Q;
T2:=LT16EQ;
T3:=TON_02.Q;
T4:=( (LT16LOG >= NIVELL_A-16 ) AND (LT16LOG <= NIVELL_A+16 ) );

```

```

//_____

// EVOLUCIO D ESTATS

CASE ESTAT OF
  0:

      IF (T0 = TRUE ) THEN
          ESTAT:=1;
      END_IF;

  1:

      IF (T1 = TRUE ) THEN
          ESTAT:=2;
      END_IF;

  2:

      IF (T2 = TRUE ) THEN
          ESTAT:=3;
      END_IF;

  3:

      IF (T3 = TRUE ) THEN
          ESTAT:=4;
      END_IF;

  4:

      IF (T4 = TRUE ) THEN
          ESTAT:=0;
      END_IF;

END_CASE;

//_____

//ACTUALITZACIÓ D ESTATS

E0:= FALSE;
E1:= FALSE;
E2:= FALSE;
E3:= FALSE;
E4:= FALSE;

IF (ESTAT = 0) THEN
    E0 := TRUE;
ELSIF (ESTAT = 1) THEN
    E1 := TRUE;
ELSIF (ESTAT = 2) THEN
    E2 := TRUE;
ELSIF (ESTAT = 3) THEN
    E3 := TRUE;
ELSIF (ESTAT = 4) THEN
    E4 := TRUE;
ELSE
END_IF;

//_____

//TEMPORITZADORS
TON_01(IN:=E1, PT:=T#1s);
TON_02(IN:=E3, PT:=T#40s);
TON_05(IN:=COUNT, PT:=T#2s); //TEMPS DE DETECCIÓ DE RECIRCULACIÓ

//_____

//SORTIDES

```

```























V11:= E1 OR E2 OR E3;
V12:= E1 OR E2 OR E3 OR E4;
V13:= E4;
VP1:= E2 OR E3;
P1:=BOOL_TO_INT(VP1)*32500;

```

```
END_PROGRAM
```

10.1.3. Programa 2

- Variables locals:

Name	Type
 TON_05	TON
 COUNT	BOOL
 E1	BOOL
 E2	BOOL
 E3	BOOL
 E4	BOOL
 LT16A	UDINT
 LT16A2	UDINT
 LT16B	INT
 LT16EQ	BOOL
 LT16LOG	INT
 N	INT
 NIVELL_A	INT
 PRIMER	INT
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 T4	BOOL
 TON_01	TON
 TON_02	TON
 VP1	BOOL

- Programa:

```

PROGRAM _INIT
    ESTAT:=0;
    E0 := TRUE;
    E1 := FALSE;
    E2 := FALSE;
    E3 := FALSE;
    E4 := FALSE;
    LT16A:=0;
    LT16B:=0;
    COUNT:=0;
    N:=29;
    PRIMER := 0;
    V11:=FALSE;
    V12:=FALSE;
    V13:=FALSE;
    V14:=FALSE;
    V15:=0;
    P1:=0;
END_PROGRAM

```



```
PROGRAM _CYCLIC
```

```
//CONTROL DE VARIABLES
//LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
LT16LOG := 32767 - LT16;
```

```
//CONTROL DE LT16EQ
```

```
IF (E2=TRUE) THEN
    LT16EQ := FALSE;
    COUNT := TRUE;

    IF (TON_05.Q = TRUE) THEN
        LT16A := LT16LOG;
        COUNT := FALSE;

        IF (PRIMER = 0) THEN
            LT16A2 := LT16A;
            LT16A := 0;
            PRIMER := 1;

            ELSIF (PRIMER = 1) THEN

                IF (LT16A >= LT16A2) THEN
                    LT16EQ := TRUE;
                    LT16A2:= LT16A;
                    PRIMER := 0;
                    LT16A := 0;

                ELSE
                    LT16A2 := LT16A;
                END_IF;

            ELSE
                END_IF;
        END_IF;
    ELSE
        COUNT := FALSE;
    END_IF;
```

```
//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)
```

```
IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;
```

```
//_____
```

```
// TRANSICIONS ENTRADES
```

```
T0:= START AND LT5 AND LT4 AND LT3;
```

```
T1:=TON_01.Q;
```

```
T2:=LT16EQ;
```

```
T3:=TON_02.Q;
```

```
T4:=( (LT16LOG >= NIVELL_A-16 ) AND (LT16LOG <= NIVELL_A+16 ) ) AND LT5 AND LT4 AND
```

```
LT3;
```

```
//_____
```

```
// EVOLUCIO D ESTATS
```

```
CASE ESTAT OF
    0:
```

```
    IF (T0 = TRUE ) THEN
        ESTAT:=1;
    END_IF;
```

```
1:

    IF (T1 = TRUE ) THEN
        ESTAT:=2;
    END_IF;

2:

    IF (T2 = TRUE ) THEN
        ESTAT:=3;
    END_IF;

3:

    IF (T3 = TRUE ) THEN
        ESTAT:=4;
    END_IF;

4:

    IF (T4 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

END_CASE;

//_____

//ACTUALITZACIÓ D ESTATS

E0:= FALSE;
E1:= FALSE;
E2:= FALSE;
E3:= FALSE;
E4:= FALSE;

IF (ESTAT = 0) THEN
    E0 := TRUE;
ELSIF (ESTAT = 1) THEN
    E1 := TRUE;
ELSIF (ESTAT = 2) THEN
    E2 := TRUE;
ELSIF (ESTAT = 3) THEN
    E3 := TRUE;
ELSIF (ESTAT = 4) THEN
    E4 := TRUE;
ELSE
    END_IF;

//_____

//TEMPORITZADORS
TON_01(IN:=E1, PT:=T#1s);
TON_02(IN:=E3, PT:=T#40s);
TON_05(IN:=COUNT, PT:=T#2s); //TEMPS DE DETECCIÓ DE RECIRCULACIÓ





























//_____

//SORTIDES
V11:= E1 OR E2 OR E3;
V12:= E1 OR E2 OR E3 OR E4;
V13:= E1 OR E4;
VP1:= E2 OR E3;
P1:=BOOL_TO_INT(VP1)*32500;

END_PROGRAM
```

10.1.4. Programa 3

- Variables locals:

Name ▲	Type
 COUNT	BOOL
 E1	BOOL
 E2	BOOL
 E3	BOOL
 E4	BOOL
 E5	BOOL
 LT16A	UDINT
 LT16A2	UDINT
 LT16B	INT
 LT16EQ	BOOL
 LT16LOG	INT
 N	INT
 NIVELL_A	INT
 PRIMER	INT
 T0	BOOL
 T1	BOOL
 T2A	BOOL
 T2B	BOOL
 T2C	BOOL
 T3	BOOL
 T4	BOOL
 T5	BOOL
 T6	BOOL
 TON_01	TON
 TON_02	TON
 TON_05	TON
 VP1	BOOL
 WV15	BOOL

- Programa:

```

PROGRAM _INIT
    ESTAT:=0;
    E0 := TRUE;
    E1 := FALSE;
    E2 := FALSE;
    E3 := FALSE;
    E4 := FALSE;
    E5 := FALSE;
    E6 := FALSE;
    LT16A:=0;
    LT16B:=0;
    COUNT:=0;
    N:=19;
    PRIMER := 0;
    V11:=FALSE;
    V12:=FALSE;
    V13:=FALSE;
    V14:=FALSE;
    V15:=0;
    P1:=0;

END_PROGRAM

PROGRAM _CYCLIC

    //CONTROL DE VARIABLES

```

```

        //LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
LT16LOG := 32767 - LT16;

//CONTROL DE LT16EQ

IF (E2=TRUE) THEN
    LT16EQ := FALSE;
    COUNT := TRUE;

    IF (TON_05.Q = TRUE) THEN
        LT16A := LT16LOG;
        COUNT := FALSE;

        IF (PRIMER = 0) THEN
            LT16A2 := LT16A;
            LT16A := 0;
            PRIMER := 1;

        ELSIF (PRIMER = 1) THEN

            IF (LT16A >= LT16A2) THEN
                LT16EQ := TRUE;
                LT16A2:= LT16A;
                PRIMER := 0;
                LT16A := 0;

            ELSE
                LT16A2 := LT16A;
            END_IF;

        ELSE
            END_IF;

    END_IF;
ELSE
    COUNT := FALSE;
END_IF;

//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)

IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;

// _____

// TRANSICIONS ENTRADES
T0:= START AND LT5 AND LT4 AND LT3;
T1:=TON_01.Q;
T2A:=LT16EQ AND LT3;
T2B:= NOT(LT3) AND LT1 AND NOT(LT16EQ);
T2C:= NOT(LT3) AND NOT(LT1) AND NOT(LT16EQ);
T3:= TON_02.Q AND LT3;
T4:=( (LT16LOG >= NIVELL_A-16 ) AND (LT16LOG <= NIVELL_A+16 ) ) AND LT5 AND LT4 AND
LT3;
T5:= NOT(LT1) OR LT5;
T6:= TON_03.Q; //REARME

// _____

// EVOLUCIO D ESTATS

CASE ESTAT OF
    0:

        IF (T0 = TRUE ) THEN
            ESTAT:=1;
        END_IF;

```

```

1:

    IF ( T1 = TRUE ) THEN
        ESTAT:=2;
    END_IF;

2:

    IF ( T2A = TRUE ) THEN
        ESTAT:=3;
    ELSIF ( T2B = TRUE ) THEN
        ESTAT:=5;
    ELSIF ( T2C = TRUE ) THEN
        ESTAT:=6;
    END_IF;

3:

    IF ( T3 = TRUE ) THEN
        ESTAT:=4;
    END_IF;

4:

    IF ( T4 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

5:

    IF ( T5 = TRUE ) THEN
        ESTAT:=1;
    END_IF;

6:

    IF ( T6 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

END_CASE;

// _____

//ACTUALITZACIÓ D ESTATS

E0:= FALSE;
E1:= FALSE;
E2:= FALSE;
E3:= FALSE;
E4:= FALSE;
E5:= FALSE;
E6:= FALSE;

IF (ESTAT = 0) THEN
    E0 := TRUE;
ELSIF (ESTAT = 1) THEN
    E1 := TRUE;
ELSIF (ESTAT = 2) THEN
    E2 := TRUE;
ELSIF (ESTAT = 3) THEN
    E3 := TRUE;
ELSIF (ESTAT = 4) THEN
    E4 := TRUE;
ELSIF (ESTAT = 5) THEN
    E5 := TRUE;
ELSIF (ESTAT = 6) THEN
    E6 := TRUE;
ELSE
END_IF;

```

```
// _____

//TEMPORITZADORS
TON_01(IN:=E1, PT:=T#1s);           //TEMPS DE TRANSICIÓ
TON_02(IN:=E3, PT:=T#40s);          //RECIRCULACIÓ
TON_03(IN:=NOT(STOP), PT:=T#5s);    //REARME
TON_05(IN:=COUNT, PT:=T#2s);       //TEMPS DE DETECCIÓ DE RECIRCULACIÓ

































// _____

//SORTIDES
V11:= E1 OR E2 OR E3;
V12:= E1 OR E2 OR E3 OR E4;
V13:= E4;
VP1:= E2 OR E3;
P1:=BOOL_TO_INT(VP1)*32500;
VV15:= E5;
V15:=BOOL_TO_INT(VV15)*32500;
ALARMA:= E6;

END_PROGRAM
```

10.1.5. Programa 4

- Variables locals:

Name	Type
 TON_05	TON
 COUNT	BOOL
 E1	BOOL
 E2	BOOL
 E3	BOOL
 E4	BOOL
 E5	BOOL
 LT16A	UDINT
 LT16A2	UDINT
 LT16B	INT
 LT16EQ	BOOL
 LT16LOG	INT
 N	INT
 NIVELL_A	INT
 N_FUGA	INT
 PERD	BOOL
 PRIMER	INT
 T0	BOOL
 T1	BOOL
 T2A	BOOL
 T2B	BOOL
 T2C	BOOL
 T3B	BOOL
 T3A	BOOL
 T4	BOOL
 T5	BOOL
 T6	BOOL
 T7	BOOL
 TON_01	TON
 TON_02	TON
 VP1	BOOL
 VV15	BOOL

- Programa:

```

PROGRAM _INIT
  ESTAT:=0;
  E0 := TRUE;
  E1 := FALSE;
  E2 := FALSE;
  E3 := FALSE;
  E4 := FALSE;
  E5 := FALSE;
  E6 := FALSE;
  E7 := FALSE;
  LT16A:=0;
  LT16A2:=0;
  N:=29;
  N_FUGA:=0;
  PRIMER := 0;
  V11:=FALSE;
  V12:=FALSE;
  V13:=FALSE;
  V14:=FALSE;
  V15:=0;
  P1:=0;
  LT16LOG:=0;
  ALARMA := FALSE;
  PERD := FALSE;
END_PROGRAM

PROGRAM _CYCLIC

  //CONTROL DE VARIABLES

  //LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
  LT16LOG := 32767 - LT16;

  //CONTROL DE LT16EQ

  IF (E2=TRUE) THEN
    LT16EQ := FALSE;
    COUNT := TRUE;

    IF (TON_05.Q = TRUE) THEN
      LT16A := LT16LOG;
      COUNT := FALSE;

      IF (PRIMER = 0) THEN
        LT16A2 := LT16A;
        LT16A := 0;
        PRIMER := 1;

        ELSIF (PRIMER = 1) THEN

          IF (LT16A >= LT16A2) THEN
            LT16EQ := TRUE;
            LT16A2:= LT16A;
            PRIMER := 0;
            LT16A := 0;

          ELSE
            LT16A2 := LT16A;
          END_IF;

        ELSE
          END_IF;

      END_IF;

    ELSE
      COUNT := FALSE;
    END_IF;
  
```

```

// CONTROL DE FUGES

IF (E3=TRUE AND (LT16LOG < (LT16A2 - 300))) THEN //300 ÉS LA SENSIBILITAT DE LA
FUGA (ATENCIÓ!!: SI ES POSA MOLT BAIXA DETECTA LES OSCIL·LACIONS DE L'AIGUA)

    IF (N_FUGA < 5) THEN
        N_FUGA := N_FUGA + 1;

    ELSIF (N_FUGA = 5) THEN
        PERD := TRUE;
        N_FUGA := 0;
    END_IF;

END_IF;

//ALARMA_BLINK
IF (ALARMA_BLINK = TRUE) THEN
    TCOUNT := TCOUNT + 1;
    IF (TCOUNT =19) THEN
        ALARMA := NOT(ALARMA);
        TCOUNT:=0;
    ELSE
        END_IF;
    ELSIF (NALARMA = TRUE) THEN
        ALARMA := TRUE;
    ELSE
        ALARMA := FALSE;
        TCOUNT:=0;
    END_IF;

//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)

IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;

// _____

// TRANSICIONS ENTRADES
T0:= START AND LT5 AND LT4 AND LT3;
T1:=TON_01.Q;
T2A:=LT16EQ AND LT3;
T2B:= NOT(LT3) AND LT1 AND NOT(LT16EQ);
T2C:= NOT(LT3) AND NOT(LT1) AND NOT(LT16EQ);
T3A:= TON_02.Q AND NOT(PERD);
T3B:= PERD;
T4:=(LT16LOG >= NIVELL_A-16 ) AND (LT16LOG <= NIVELL_A+16 ) AND LT5 AND LT4 AND
LT3;
T5:= NOT(LT1) OR LT5;
T6:= TON_03.Q; //REARME
T7:= TON_03.Q; //REARME

// _____

// EVOLUCIO D ESTATS

CASE ESTAT OF
    0:

        IF (T0 = TRUE ) THEN
            ESTAT:=1;
        END_IF;

    1:

        IF (T1 = TRUE ) THEN
            ESTAT:=2;
        END_IF;

```



```

2:

    IF ( T2A = TRUE ) THEN
        ESTAT:=3;
    ELSIF ( T2B = TRUE ) THEN
        ESTAT:=5;
    ELSIF ( T2C = TRUE ) THEN
        ESTAT:=6;
    END_IF;

3:

    IF ( T3A = TRUE ) THEN
        ESTAT:=4;
    ELSIF ( T3B = TRUE ) THEN
        ESTAT:=7;
    END_IF;

4:

    IF ( T4 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

5:

    IF ( T5 = TRUE ) THEN
        ESTAT:=1;
    END_IF;

6:

    IF ( T6 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

7:

    IF ( T7 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

END_CASE;

```

```
//ACTUALITZACIÓ D ESTATS
```

```

E0:= FALSE;
E1:= FALSE;
E2:= FALSE;
E3:= FALSE;
E4:= FALSE;
E5:= FALSE;
E6:= FALSE;
E7:= FALSE;

IF (ESTAT = 0) THEN
    E0 := TRUE;
ELSIF (ESTAT = 1) THEN
    E1 := TRUE;
ELSIF (ESTAT = 2) THEN
    E2 := TRUE;
ELSIF (ESTAT = 3) THEN
    E3 := TRUE;
ELSIF (ESTAT = 4) THEN
    E4 := TRUE;
ELSIF (ESTAT = 5) THEN
    E5 := TRUE;

```

```

ELSIF (ESTAT = 6) THEN
    E6 := TRUE;
ELSIF (ESTAT = 7) THEN
    E7 := TRUE;
ELSE
END_IF

// _____

//TEMPORITZADORS
TON_01(IN:=E1, PT:=T#1s);           //TEMPS DE TRANSICIÓ
TON_02(IN:=E3, PT:=T#40s);         //RECIRCULACIÓ
TON_03(IN:=NOT(STOP), PT:=T#5s);   //REARME
TON_05(IN:=COUNT, PT:=T#2s);     //TEMPS DE DETECCIÓ DE RECIRCULACIÓ

// _____
























//SORTIDES
V11:= E1 OR E2 OR E3;
V12:= E1 OR E2 OR E3 OR E4;
V13:= E4;
VP1:= E2 OR E3;
P1:=BOOL_TO_INT(VP1)*32500;
VV15:= E5;
V15:=BOOL_TO_INT(VV15)*32500;
NALARMA:= E6;
ALARMA_BLINK := E7;





















































END_PROGRAM

```

10.1.6. Programa 5

- Variables locals:

Name	Type
 LT13	INT
 TMOSTREIG	REAL
 SENSIB	INT
 LT16A2	UDINT
 LT16A	UDINT
 TON_02	TON
 TON_01	TON
 TON_05	TON
 N	INT
 LT16B	INT
 LT16LOG	INT
 NIVELL_A	INT
 N_FUGA	INT
 CONS_NVAS	INT
 PRIMER	INT
 T51	BOOL
 T5	BOOL
 T71	BOOL
 T70	BOOL
 T61	BOOL
 T60	BOOL
 T30	BOOL
 T50	BOOL

 VP1	BOOL
 T41	BOOL
 T31	BOOL
 T3B	BOOL
 T4	BOOL
 T0	BOOL
 T3A	BOOL
 T2C	BOOL
 T2B	BOOL
 T2A	BOOL
 T1	BOOL
 T20	BOOL
 r6	BOOL
 r5	BOOL
 r4	BOOL
 r3	BOOL
 r2	BOOL
 r1	BOOL
 PERD	BOOL
 OV15	BOOL
 T21	BOOL
 COUNT	BOOL
 T7	BOOL
 T6	BOOL
 E71	BOOL
 E102	BOOL
 E104	BOOL
 VV15	BOOL
 LT16EQ	BOOL
 E61	BOOL
 OP1	BOOL
 E70	BOOL
 E50	BOOL
 E60	BOOL
 E51	BOOL
 OV12	BOOL
 E41	BOOL
 E40	BOOL
 E31	BOOL
 E30	BOOL
 E21	BOOL
 E20	BOOL
 E5	BOOL
 E4	BOOL
 E3	BOOL
 E2	BOOL
 E1	BOOL
 OV11	BOOL
 OV14	BOOL
 OV13	BOOL
 T40	BOOL
 CI	BOOL

- Programa:

```
PROGRAM _INIT
```

```

    LT16A:=0;
    LT16B:=0;
    COUNT:=0;
    N_FUGA:=0;
    PRIMER := 0;
    V11:=FALSE;
```

```

V12:=FALSE;
V13:=FALSE;
V14:=FALSE;
V15:=0;
P1:=0;
LT16LOG:=0;
ALARMA := FALSE;
PERD := FALSE;

//INICIALITZACIÓ DESTATS
ESTAT:= 0;
GRAF_V11:= 20;
GRAF_V12:= 30;
GRAF_V13:= 40;
GRAF_V14:= 50;
GRAF_V15:= 60;
GRAF_P1:= 70;
GEMMA:= 100;
TECLAT:= 110;

//VARIABLES DE CONTROL
SENSIB:= 300;
TMOSTREIG:= 2.0;

//BOTONS MANUAL
OV11:=FALSE;
OV12:=FALSE;
OV13:=FALSE;
OV14:=FALSE;
OV15:=FALSE;
OP1:=FALSE;

END_PROGRAM

PROGRAM _CYCLIC
//CONTROL DE VARIABLES

//LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
LT16LOG := 32767 - LT16;

//CONTROL DE LT16EQ

IF (E2=TRUE) THEN
    LT16EQ := FALSE;
    COUNT := TRUE;

    IF (TON_05.Q = TRUE) THEN
        LT16A := LT16LOG;
        COUNT := FALSE;

        IF (PRIMER = 0) THEN
            LT16A2 := LT16A;
            LT16A := 0;
            PRIMER := 1;

        ELSIF (PRIMER = 1) THEN

            IF (LT16A >= LT16A2) THEN
                LT16EQ := TRUE;
                LT16A2:= LT16A;
                PRIMER := 0;
                LT16A := 0;

            ELSE
                LT16A2 := LT16A;
            END_IF;

        ELSE
            END_IF;

    END_IF;

ELSE
    END_IF;

```

```

COUNT := FALSE;
END_IF;

// CONTROL DE FUGES

IF (E3=TRUE AND (LT16LOG < (LT16A2 - SENSIB))) THEN //300 ÉS LA SENSIBILITAT DE LA
FUGA (ATENCIÓ!!!: SI ES POSA MOLT BAIXA DETECTA LES OSCIL·LACIONS DE L'AIGUA)

    IF (N_FUGA < 5) THEN
        N_FUGA := N_FUGA + 1;

    ELSIF (N_FUGA = 5) THEN
        PERD := TRUE;
        N_FUGA := 0;
    END_IF;

END_IF;

//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)

IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;

// _____

// TRANSICIONS ENTRADES

//G0

T0:= E102 AND LT5 AND LT4 AND LT3;
T1:=TON_01.Q;
T2A:=LT16EQ AND LT3;
T2B:= NOT(LT3) AND LT1 AND NOT(LT16EQ);
T2C:= NOT(LT3) AND NOT(LT1) AND NOT(LT16EQ);
T3A:= TON_02.Q AND NOT(PERD);
T3B:= PERD;
T4:=( (LT16LOG >= NIVELL_A-16 ) AND (LT16LOG <= NIVELL_A+16 ) ) AND LT5 AND LT4 AND
LT3;
T5:= NOT(LT1) OR LT5;
//T6:= TON_03.Q; //REARME
//T7:= TON_03.Q; //REARME

//G20 V11

T20:= (E104 AND OV11)OR E1 OR E2 OR E3;
T21:= (E104 AND NOT(OV11))OR E4 OR E5 OR E6 OR E7 OR E101 ;

//G30 V12

T30:= (E104 AND OV12)OR E1 OR E2 OR E3 OR E4 OR E101;
T31:= (E104 AND NOT(OV12))OR (E0 AND NOT(E101) AND NOT(E104)) OR E5 OR E6 OR E7 OR
E100;

//G40 V13

T40:= (E104 AND OV13)OR E4 OR E101;
T41:= (E104 AND NOT(OV13))OR (E0 AND NOT(E101) AND NOT(E104)) OR E100;

//G50 V14

T50:= (E104 AND OV14);
T51:= (E104 AND NOT(OV14)) OR E101;

//G60 V15

T60:= (E104 AND OV15) OR E5 OR E101;
T61:= (E104 AND NOT(OV15)) OR E1 OR E100 ;

//G70 P1

```

```

T70:= (E104 AND OP1) OR E2 OR E3;
T71:= (E104 AND NOT(OP1)) OR E4 OR E5 OR E6 OR E7 OR E101;

//G100 GEMMA

CI:= LT5 AND NOT(VP1) AND NOT(VV15) AND NOT(V11) AND NOT(V12) AND NOT(V13) AND
NOT(V14);
//VSTART i VSTOP SÓN VARIABLES DEL SISTEMA SCADA
r1:= (VSTART OR START) AND NOT(A_M) AND E121 AND CI;
r2:= (VSTART OR START) AND A_M AND E121;
r3:= NOT(STOP) OR VSTOP AND NOT(A_M);
r4:= NOT(STOP) OR VSTOP;
r5:= E0;
r6:= (VSTART OR START) AND NOT(A_M) AND E121 AND NOT(CI);
r7:= LT3 AND LT4 AND LT5;

//_____

// EVOLUCIO D ESTATS

//GRAF GENERAL G0
CASE ESTAT OF
  0:

      IF (T0 = TRUE ) THEN
          ESTAT:=1;
      END_IF;

  1:

      IF (T1 = TRUE ) THEN
          ESTAT:=2;
      END_IF;

  2:

      IF (T2A = TRUE ) THEN
          ESTAT:=3;
      ELSIF (T2B = TRUE) THEN
          ESTAT:=5;
      ELSIF (T2C = TRUE) THEN
          ESTAT:=6;
      END_IF;

  3:

      IF (T3A = TRUE ) THEN
          ESTAT:=4;
      ELSIF (T3B = TRUE) THEN
          ESTAT:=7;
      END_IF;

  4:

      IF (T4 = TRUE ) THEN
          ESTAT:=0;
      END_IF;

  5:

      IF (T5 = TRUE ) THEN
          ESTAT:=1;
      END_IF;

  6:

      IF (T6 = TRUE ) THEN
          ESTAT:=0;
      END_IF;

```

```
7:

    IF ( T7 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

END_CASE;

//GRAF G20
CASE GRAF_V11 OF
20:

    IF ( T20 = TRUE ) THEN
        GRAF_V11:=21;
    END_IF;

21:

    IF ( T21 = TRUE ) THEN
        GRAF_V11:=20;
    END_IF;

END_CASE;

//GRAF G30
CASE GRAF_V12 OF
30:

    IF ( T30 = TRUE ) THEN
        GRAF_V12:=31;
    END_IF;

31:

    IF ( T31 = TRUE ) THEN
        GRAF_V12:=30;
    END_IF;

END_CASE;

//GRAF G40
CASE GRAF_V13 OF
40:

    IF ( T40 = TRUE ) THEN
        GRAF_V13:=41;
    END_IF;

41:

    IF ( T41 = TRUE ) THEN
        GRAF_V13:=40;
    END_IF;

END_CASE;

//GRAF G50
CASE GRAF_V14 OF
50:

    IF ( T50 = TRUE ) THEN
        GRAF_V14:=51;
    END_IF;

51:

    IF ( T51 = TRUE ) THEN
        GRAF_V14:=50;
    END_IF;

END_CASE;
```

```
//GRAF G60
CASE GRAF_V15 OF
  60:

    IF (T60 = TRUE ) THEN
      GRAF_V15:=61;
    END_IF;

  61:

    IF (T61 = TRUE ) THEN
      GRAF_V15:=60;
    END_IF;

END_CASE;

//GRAF G70
CASE GRAF_P1 OF
  70:

    IF (T70 = TRUE ) THEN
      GRAF_P1:=71;
    END_IF;

  71:

    IF (T71 = TRUE ) THEN
      GRAF_P1:=70;
    END_IF;

END_CASE;

//GRAF G100
CASE GEMMA OF
  100:

    IF (r6 = TRUE ) THEN
      GEMMA:=101;
    ELSIF (r1 = TRUE) THEN
      GEMMA:=102;
    ELSIF (r2 = TRUE) THEN
      GEMMA:=104;
    END_IF;

  101:

    IF (r7 = TRUE ) THEN
      GEMMA:=100;
    END_IF;

  102:

    IF (r4 = TRUE ) THEN
      GEMMA:=103;
    END_IF;

  103:

    IF (r5 = TRUE ) THEN
      GEMMA:=100;
    END_IF;

  104:

    IF (r3 = TRUE ) THEN
      GEMMA:=101;
    END_IF;

END_CASE;
```



```
// _____  
  
//ACTUALITZACIÓ D ESTATS  
  
//GRAF G0 (ESTAT)  
  
E0:= FALSE;  
E1:= FALSE;  
E2:= FALSE;  
E3:= FALSE;  
E4:= FALSE;  
E5:= FALSE;  
E6:= FALSE;  
E7:= FALSE;  
  
IF (ESTAT = 0) THEN  
    E0 := TRUE;  
ELSIF (ESTAT = 1) THEN  
    E1 := TRUE;  
ELSIF (ESTAT = 2) THEN  
    E2 := TRUE;  
ELSIF (ESTAT = 3) THEN  
    E3 := TRUE;  
ELSIF (ESTAT = 4) THEN  
    E4 := TRUE;  
ELSIF (ESTAT = 5) THEN  
    E5 := TRUE;  
ELSIF (ESTAT = 6) THEN  
    E6 := TRUE;  
ELSIF (ESTAT = 7) THEN  
    E7 := TRUE;  
ELSE  
END_IF  
  
//GRAF G20 (GRAF_V11)  
  
E20:= FALSE;  
E21:= FALSE;  
  
IF (GRAF_V11 = 20) THEN  
    E20 := TRUE;  
ELSIF (GRAF_V11 = 21) THEN  
    E21 := TRUE;  
ELSE  
END_IF  
  
//GRAF G30 (GRAF_V12)  
  
E30:= FALSE;  
E31:= FALSE;  
  
IF (GRAF_V12 = 30) THEN  
    E30 := TRUE;  
ELSIF (GRAF_V12 = 31) THEN  
    E31 := TRUE;  
ELSE  
END_IF  
  
//GRAF G40 (GRAF_V13)  
  
E40:= FALSE;  
E41:= FALSE;  
  
IF (GRAF_V13 = 40) THEN  
    E40 := TRUE;  
ELSIF (GRAF_V13 = 41) THEN  
    E41 := TRUE;  
ELSE  
END_IF  
  
//GRAF G50 (GRAF_V14)
```

```

E50:= FALSE;
E51:= FALSE;

IF (GRAF_V14 = 50) THEN
    E50 := TRUE;
ELSIF (GRAF_V14 = 51) THEN
    E51 := TRUE;
ELSE
    END_IF

//GRAF G60 (GRAF_V15)

E60:= FALSE;
E61:= FALSE;

IF (GRAF_V15 = 60) THEN
    E60 := TRUE;
ELSIF (GRAF_V15 = 61) THEN
    E61 := TRUE;
ELSE
    END_IF

//GRAF G70 (GRAF_P1)

E70:= FALSE;
E71:= FALSE;

IF (GRAF_P1 = 70) THEN
    E70 := TRUE;
ELSIF (GRAF_P1 = 71) THEN
    E71 := TRUE;
ELSE
    END_IF

//GRAF G100 (GEMMA)

E100:= FALSE;
E101:= FALSE;
E102:= FALSE;
E103:= FALSE;
E104:= FALSE;

IF (GEMMA = 100) THEN
    E100 := TRUE;
ELSIF (GEMMA = 101) THEN
    E101 := TRUE;
ELSIF (GEMMA = 102) THEN
    E102 := TRUE;
ELSIF (GEMMA = 103) THEN
    E103 := TRUE;
ELSIF (GEMMA = 104) THEN
    E104 := TRUE;
ELSE
    END_IF

//_____

//TEMPORITZADORS
TON_01(IN:=E1, PT:=T#1s);           //TEMPS DE TRANSICIÓ
TON_02(IN:=E3, PT:=T#40s);          //RECIRCULACIÓ
TON_05(IN:=COUNT, PT:=REAL_TO_TIME(TMOSTREIG*1000.0)); //TEMPS DE DETECCIÓ DE
RECIRCULACIÓ

//_____

//SORTIDES
V11:= E21;
V12:= E31;
V13:= E41;
V14:= E51;
VP1:= E71;

```

```


P1:=BOOL_TO_INT(VP1)*32500;
VV15:= E61;
V15:=BOOL_TO_INT(VV15)*32500;

//LLUMS
LT13:= BOOL_TO_INT(LT3)*10;

```

END_PROGRAM

- Variables locals del programa d'aturada d'Emergència:

Name	Type
 T121B	BOOL
 T121A	BOOL

- Programa d'aturada d'Emergència:

```

PROGRAM _INIT
    EMERGENCIA:= 120;
END_PROGRAM

```

PROGRAM _CYCLIC

```

//CONTROL DE VARIABLES

//FORÇAT DE LES ETAPES
IF (E122 = TRUE) THEN
    ESTAT:= 0;
    GRAF_V11:= 20;
    GRAF_V12:= 30;
    GRAF_V13:= 40;
    GRAF_V14:= 50;
    GRAF_V15:= 60;
    GRAF_P1:= 70;
    GEMMA:= 100;
    TECLAT:= 110;
ELSE
END_IF;

IF (E124 = TRUE) THEN
    GEMMA:= 101;
ELSE
END_IF;

//ALARMA_BLINK
IF (ALARMA_BLINK = TRUE) THEN
    TCOUNT := TCOUNT + 1;
    IF (TCOUNT =99) THEN
        ALARMA := NOT(ALARMA);
        TCOUNT:=0;
    ELSE
    END_IF;
ELSIF (NALARMA = TRUE) THEN
    ALARMA := TRUE;
ELSE
    ALARMA := FALSE;
    TCOUNT:=0;
END_IF;

//
// TRANSICIONS ENTRADES
T120:= OnOff;
T121A:= (E101 AND r7) OR (E103 AND E0) OR NOT(OnOff);

```

```

T121B:= NOT(PE) OR E6 OR E7;
T122:= PE;
T123:= TON_03.Q;
T124:= E100;

// _____

// EVOLUCIO D ESTATS

CASE EMERGENCIA OF
  120:

      IF (T120 = TRUE ) THEN
          EMERGENCIA:=121;
      END_IF;

  121:

      IF (T121A = TRUE ) THEN
          EMERGENCIA:=120;
      ELSIF (T121B = TRUE ) THEN
          EMERGENCIA:=122;
      END_IF;

  122:

      IF (T122 = TRUE ) THEN
          EMERGENCIA:=123;
      END_IF;

  123:

      IF (T123 = TRUE ) THEN
          EMERGENCIA:=124;
      END_IF;

  124:

      IF (T124 = TRUE ) THEN
          EMERGENCIA:=120;
      END_IF;

END_CASE;

// _____

//ACTUALITZACIÓ D ESTATS

E120:= FALSE;
E121:= FALSE;
E122:= FALSE;
E123:= FALSE;
E124:= FALSE;

IF (EMERGENCIA = 120) THEN
    E120 := TRUE;
ELSIF (EMERGENCIA = 121) THEN
    E121 := TRUE;
ELSIF (EMERGENCIA = 122) THEN
    E122 := TRUE;
ELSIF (EMERGENCIA = 123) THEN
    E123 := TRUE;
ELSIF (EMERGENCIA = 124) THEN
    E124 := TRUE;
ELSE
    END_IF;

// _____
//TEMPORITZADORS

TON_03(IN:=((STOP=FALSE) OR (VSTOP=FALSE)), PT:=T#5s); //REARME ON VSTOP















































```

























ES VARIABLES DEL SISTEMA SCADA

```
//  
//SORTIDES  
NALARMA:= E122;  
ALARMA_BLINK:= E123;  
POWER:= E121;  
  
END_PROGRAM
```

10.1.7. Programa 1 amb control PID

- Variables locals:

Name	Type
 T15	BOOL
 T12C	BOOL
 YV	REAL
 Y	REAL
 V_PID	BOOL
 WV15	BOOL
 VP1	BOOL
 UP	REAL
 UI_1	REAL
 UI	REAL
 UD	REAL
 TON_02	TON
 TON_01	TON
 TI	REAL
 TEMPS	REAL
 TD	REAL
 T112	BOOL
 T111	BOOL
 T110	BOOL
 T22	BOOL
 T21	BOOL
 T20	BOOL
 T14	BOOL
 T13	BOOL
 T12B	BOOL
 T12A	BOOL
 T11	BOOL
 T10	BOOL
 T6	BOOL
 T5	BOOL
 T4	BOOL
 T3	BOOL
 T2B	BOOL
 T2A	BOOL
 T1	BOOL
 T0	BOOL
 PT11V_1	REAL
 PT11V	REAL
 PRIMER	INT
 N_VAS	BOOL
 N_PT11	BOOL
 N_INIT	BOOL
 NIVELL_A	INT
 N	INT
 LT16LOG	INT
 LT16B	INT
LT16A	INT

	KP	REAL
	ERR	REAL
	ED_1	REAL
	ED	REAL
	E112	BOOL
	E111	BOOL
	E110	BOOL
	E21	BOOL
	E20	BOOL
	E13	BOOL
	E12	BOOL
	E11	BOOL
	E10	BOOL
	E5	BOOL
	E4	BOOL
	E3	BOOL
	E2	BOOL
	E1	BOOL
	COUNT	INT
	CONS_NVAS	INT
	CONS_NPT11	REAL
	CONS_NINIT	INT
	c	REAL
	b	REAL

- Programa:

PROGRAM _INIT

```

    LT16A:=0;
    LT16B:=0;
    COUNT:=0;
    PRIMER := 0;
    V11:=FALSE;
    V12:=FALSE;
    V13:=FALSE;
    V14:=FALSE;
    V15:=0;
    P1:=0;
    ED:=0;
    ED_1:=0;
    PT11V:=0;
    PT11V_1:=0;
    UI:=0;
    UD:=0;
    UP:=0;

    //VARIABLES DE CONTROL

    CONS_NPT11:= 5.0;
    CONS_NVAS:=2000;
    CONS_NINIT:= 13500;
    N:=10;
    TEMPS:=0.05;

    //INICIALITZACIÓ D'ESTATS
    ESTAT:=0;
    GRAF_INIT:=10;
    GRAF_FI:=20;
    TECLAT:=110;

    //PID

    //(es considera el cyclic del programa, en aquest cas 50ms)

    TEMPS:= 0.05; //segons
    TD:=1.45;

```

```

TI:=2.6;
KP:= 0.5;
b:=1;
c:=0;

( *
PID:= (0);

PID.Enable:=1;

PID.PIDParameters.Gain:= 0.8;
PID.PIDParameters.IntegrationTime:= 2.6;
PID.PIDParameters.DerivativeTime:= 1.45;
PID.PIDParameters.FilterTime:= 0.145;

PID.MinOut:= 0;
PID.MaxOut:= 32767;
*)

END_PROGRAM

PROGRAM _CYCLIC

//CONTROL DE VARIABLES

//CONTROL DE SOBRESATURACIÓ DE CONS_NPT11
IF (CONS_NPT11 <1.44) THEN
    CONS_NPT11:= 1.44;

ELSIF (CONS_NPT11 >10) THEN
    CONS_NPT11:= 10;

END_IF;

//ALARMA_BLINK
IF (ALARMA_BLINK = TRUE) THEN
    TCOUNT := TCOUNT + 1;
    IF (TCOUNT =19) THEN
        ALARMA := NOT(ALARMA);
        TCOUNT:=0;
    ELSE
        END_IF;
    ELSE
        ALARMA := FALSE;
        TCOUNT:=0;
    END_IF;

//LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
LT16LOG := 32767 - LT16;

//PT11 PASSAT A VOLTS
PT11V:= INT_TO_REAL(PT11) * (10.0/32767.0);

//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)
IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;

//CONTROL BOMBA + PID

IF (V_PID = TRUE) THEN//Condició per iniciar el controlador PID.

    //Actualització d'estats per modificar les variables.
    TD:=TD; //Actualització de temps derivatiu.
    TI:=TI; //Actualització de temps integral.

```

```

        KP:= KP;// Actualització de constant proporcional.

        ED:= (c * CONS_NPT11 - PT11V);           //Error derivatiu.
        ERR:= CONS_NPT11 - PT11V;               //Error actual (consigna - valor del
sensor).

        UP:= KP * (b * CONS_NPT11 - PT11V); //Acció proporcional.
        UI:= UI_1 + (KP * (TEMPS/TI)*ERR); //Acció integral.
        UD:= ((TD/(TD + TEMPS*N))*UD) + (((KP*TD*N)/(TD+TEMPS*N))*(ED - ED_1));
        //Acció derivativa.

        IF (UI_1 > 10.0) AND (UI_1 < 0.0) THEN      //Limitació de l'acció integral
entre 0 i 10V.
            UI := UI_1;
        ELSE
            UI:= UI;
        END_IF;

        YV:= (UP + UI + UD); //Sortida del senyal de control de tensió de la bomba.
        Y:= (UP + UI + UD)* (32767.0/10.0); //Sortida del senyal en bits.

        PT11V_1:=PT11V;           //Assignació del valor del sensor en volts en l'ins-
tant anterior.
        ED_1:= ED;               //Assignació de l'error derivatiu en l'instant ante-
rior.
        UI_1:= UI;               //Assignació de l'acció integral en l'instant ante-
rior.

    ELSE                                //Aturada del control PID quan no es compleix la condició
V_PID
        Y:=0;
        ED:=0;
        ED_1:=0;
        PT11V:=0;
        PT11V_1:=0;
        UI:=0;
        UD:=0;
        UP:=0;

    END_IF;

    //NIVELL DE PRESSIO

    IF ((PT11V>= CONS_NPT11-0.05 ) AND (PT11V<= CONS_NPT11+0.05 )) THEN
        N_PT11 := TRUE;
    ELSE
        N_PT11 := FALSE;
    END_IF;

    //NIVELL DEL VAS

    IF (((NIVELL_A - CONS_NVAS) >= LT16LOG -16 ) AND ((NIVELL_A - CONS_NVAS) >= LT16LOG
+ 16 )) THEN
        N_VAS := TRUE;
    ELSE
        N_VAS := FALSE;
    END_IF;

    //NIVELL INICIAL

    IF ((LT16LOG >= CONS_NINIT -16) AND (LT16LOG <= CONS_NINIT +16)) THEN
        N_INIT := TRUE;
    ELSE
        N_INIT := FALSE;
    END_IF;

    //_____

    // TRANSICIONS ENTRADES

```



```

//G0

T0:= E13 AND NOT(E112) AND LT4 AND LT3;
T1:= TON_01.Q;
T2A:= N_VAS AND LT3;
T2B:= NOT(LT3);
T3:= N_PT11 ;
T4:= TON_02.Q;
T5:= (LT16LOG >= NIVELL_A -16) AND (LT16LOG <= NIVELL_A +16) AND LT4 AND LT3;
T6:= TON_03.Q; //REARME

//G10

T10:= E111 AND E0 AND LT5 AND LT4 AND LT3;
T11:= TON_01.Q;
T12A:= N_INIT AND LT4 AND LT3 AND NOT(LT2);
T12B:= LT2 AND LT3;
T12C:= NOT(LT2) AND NOT(LT3);
T13:= E22;
T14:= TON_03.Q; //REARME
T15:= TON_03.Q; //REARME

//G20

T20:= E112 AND E13 AND E0 AND NOT(LT5) AND LT4 AND LT3;
T21:= LT5;
T22:= E10;

//G110

T110:= START;
T111:= NOT(STOP) OR E6 OR E14 OR E15;
T112:= E22 OR E6 OR E14 OR E15;

// _____

// EVOLUCIO D ESTATS

//GRAF 0 GENERAL
CASE ESTAT OF
0:

    IF (T0 = TRUE ) THEN
        ESTAT:=1;
    END_IF;

1:

    IF (T1 = TRUE ) THEN
        ESTAT:=2;
    END_IF;

2:

    IF (T2A = TRUE ) THEN
        ESTAT:=3;
    ELSIF (T2B = TRUE) THEN
        ESTAT:=6;
    END_IF;

3:

    IF (T3 = TRUE ) THEN
        ESTAT:=4;
    END_IF;

4:

    IF (T4 = TRUE ) THEN
        ESTAT:=5;
    END_IF;

```

```
5:

    IF (T5 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

6:

    IF (T6 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

END_CASE;

//GRAF 10 INICI
CASE GRAF_INIT OF
10:

    IF (T10 = TRUE ) THEN
        GRAF_INIT:=11;
    END_IF;

11:

    IF (T11 = TRUE ) THEN
        GRAF_INIT:=12;
    END_IF;

12:

    IF (T12A = TRUE ) THEN
        GRAF_INIT:=13;
    ELSIF (T12B = TRUE) THEN
        GRAF_INIT:= 14;
    ELSIF (T12C = TRUE) THEN
        GRAF_INIT:= 15;
    END_IF;

13:

    IF (T13 = TRUE ) THEN
        GRAF_INIT:=10;
    END_IF;

14:

    IF (T14 = TRUE ) THEN
        GRAF_INIT:=10;
    END_IF;

15:

    IF (T15 = TRUE ) THEN
        GRAF_INIT:=10;
    END_IF;

END_CASE;

//GRAF 20 FINAL
CASE GRAF_FI OF
20:

    IF (T20 = TRUE ) THEN
        GRAF_FI:=21;
```

```

        END_IF;

21:

        IF ( T21 = TRUE ) THEN
            GRAF_FI:=22;
        END_IF;

22:

        IF ( T22 = TRUE ) THEN
            GRAF_FI:=20;
        END_IF;

END_CASE;

//GRAF 110 FINAL

CASE TECLAT OF
    110:

        IF ( T110 = TRUE ) THEN
            TECLAT:=111;
        END_IF;

    111:

        IF ( T111 = TRUE ) THEN
            TECLAT:=112;
        END_IF;

    112:

        IF ( T112 = TRUE ) THEN
            TECLAT:=110;
        END_IF;

END_CASE;

//
//ACTUALITZACIÓ D ESTATS

//GRAF GENERAL (ESTAT)
E0:= FALSE;
E1:= FALSE;
E2:= FALSE;
E3:= FALSE;
E4:= FALSE;
E5:= FALSE;
E6:= FALSE;

IF (ESTAT = 0) THEN
    E0 := TRUE;
ELSIF (ESTAT = 1) THEN
    E1 := TRUE;
ELSIF (ESTAT = 2) THEN
    E2 := TRUE;
ELSIF (ESTAT = 3) THEN
    E3 := TRUE;
ELSIF (ESTAT = 4) THEN
    E4 := TRUE;
ELSIF (ESTAT = 5) THEN
    E5 := TRUE;
ELSIF (ESTAT = 6) THEN
    E6 := TRUE;
ELSE

```

```

END_IF

//GRAF INICI (GRAF_INIT)

E10:= FALSE;
E11:= FALSE;
E12:= FALSE;
E13:= FALSE;
E14:= FALSE;
E15:= FALSE;

IF (GRAF_INIT = 10) THEN
    E10 := TRUE;
ELSIF (GRAF_INIT = 11) THEN
    E11 := TRUE;
ELSIF (GRAF_INIT = 12) THEN
    E12 := TRUE;
ELSIF (GRAF_INIT = 13) THEN
    E13 := TRUE;
ELSIF (GRAF_INIT = 14) THEN
    E14 := TRUE;
ELSIF (GRAF_INIT = 15) THEN
    E15 := TRUE;
ELSE
END_IF

//GRAF FI (GRAF_FI)

E20:= FALSE;
E21:= FALSE;
E22:= FALSE;

IF (GRAF_FI = 20) THEN
    E20 := TRUE;
ELSIF (GRAF_FI = 21) THEN
    E21 := TRUE;
ELSIF (GRAF_FI = 22) THEN
    E22 := TRUE;
ELSE
END_IF

//GRAF TECLAT (TECLAT)

E110:= FALSE;
E111:= FALSE;
E112:= FALSE;

IF (TECLAT = 110) THEN
    E110 := TRUE;
ELSIF (TECLAT = 111) THEN
    E111 := TRUE;
ELSIF (TECLAT = 112) THEN
    E112 := TRUE;
ELSE
END_IF

//_____

//TEMPORITZADORS
TON_01(IN:=E1 OR E11, PT:=T#1s);
TON_02(IN:=E4, PT:=T#40s);
TON_03(IN:=NOT(STOP), PT:=T#5s);

//TEMPS DE TRANSICIÓ
//RECIRCULACIÓ
//REARME

//_____

//SORTIDES
V11:= E1 OR E2 OR E3 OR E4;
V12:= E1 OR E2 OR E5 OR E11 OR E12;
V13:= E5 OR E11 OR E12;
V14:= E11 OR E12;
V_PID:= E3 OR E4;

```

```




































VP1:= E2 OR E12;
P1:=BOOL_TO_INT(VP1)*32500 + BOOL_TO_INT(V_PID) * REAL_TO_INT(Y);
VV15:= E21;
V15:=BOOL_TO_INT(VV15)*32500;
NALARMA:= E14;
ALARMA_BLINK := E6 OR E15;





































```

```
END_PROGRAM
```

10.1.8. Programa 2 amb control PID

- Variables locals:

Name ▲	Type
 b	REAL
 c	REAL
 CONS_NINIT	INT
 CONS_NPT11	REAL
 CONS_NVAS	INT
 COUNT	INT
 E1	BOOL
 E2	BOOL
 E3	BOOL
 E4	BOOL
 E5	BOOL
 E10	BOOL
 E11	BOOL
 E12	BOOL
 E13	BOOL
 E20	BOOL
 E21	BOOL
 E110	BOOL
 E111	BOOL
 E112	BOOL
 ED	REAL
 ED_1	REAL
 ERR	REAL
 KP	REAL
 LT16A	INT
 LT16B	INT
 LT16LOG	INT
 N	INT
 NIVELL_A	INT
 N_INIT	BOOL
 N_PT11	BOOL
 N_VAS	BOOL
 PRIMER	INT
 PT11V	REAL
 PT11V_1	REAL

	T0	BOOL
	T1	BOOL
	T2A	BOOL
	T2B	BOOL
	T3	BOOL
	T4	BOOL
	T5	BOOL
	T6	BOOL
	T10	BOOL
	T11	BOOL
	T12A	BOOL
	T12B	BOOL
	T12C	BOOL
	T13	BOOL
	T14	BOOL
	T15	BOOL
	T20	BOOL
	T21	BOOL
	T22	BOOL
	T110	BOOL
	T111	BOOL
	T112	BOOL
	TD	REAL
	TEMPS	REAL
	TI	REAL
	TON_01	TON
	TON_02	TON
	UD	REAL
	UI	REAL
	UI_1	REAL
	UP	REAL
	VP1	BOOL
	VV15	BOOL
	V_PID	BOOL
	Y	REAL
	YV	REAL

- Programa:

PROGRAM _INIT

```

    LT16A:=0;
    LT16B:=0;
    COUNT:=0;
    PRIMER := 0;
    V11:=FALSE;
    V12:=FALSE;
    V13:=FALSE;
    V14:=FALSE;
    V15:=0;
    P1:=0;
    ED:=0;
    ED_1:=0;
    PT11V:=0;
    PT11V_1:=0;
    UI:=0;
    UD:=0;
    UP:=0;

    //VARIABLES DE CONTROL

    CONS_NPT11:= 5.0;
    CONS_NVAS:=2000;
    CONS_NINIT:= 13500;
    N:=10;
    TEMPS:=0.05;

```

```

//INICIALITZACIÓ D'ESTATS
ESTAT:=0;
GRAF_INIT:=10;
GRAF_FI:=20;
TECLAT:=110;

//PID

//(es considera el cyclic del programa, en aquest cas 50ms)

TEMPS:= 0.05; //segons
TD:=1.45;
TI:=2.6;
KP:= 0.5;
b:=1;
c:=0;

( *
PID:= (0);

PID.Enable:=1;

PID.PIDParameters.Gain:= 0.8;
PID.PIDParameters.IntegrationTime:= 2.6;
PID.PIDParameters.DerivativeTime:= 1.45;
PID.PIDParameters.FilterTime:= 0.145;

PID.MinOut:= 0;
PID.MaxOut:= 32767;
*)

END_PROGRAM

PROGRAM _CYCLIC

//CONTROL DE VARIABLES

//CONTROL DE SOBRESATURACIÓ DE CONS_NPT11
IF (CONS_NPT11 <1.44) THEN
    CONS_NPT11:= 1.44;

ELSIF (CONS_NPT11 >10) THEN
    CONS_NPT11:= 10;

END_IF;

//LT16 LOGICA (LT16LOG := LT16* := +VALOR QUAN +NIVELL)
LT16LOG := 32767 - LT16;

//PT11 PASSAT A VOLTS
PT11V:= INT_TO_REAL(PT11) * (10.0/32767.0);

//MEMORITZACIÓ DE NIVELL INICIAL (NIVELL_A)
IF (E1=TRUE) THEN
    NIVELL_A:=LT16LOG;
END_IF;

//CONTROL BOMBA + PID

IF (V_PID = TRUE) THEN

    TD:=TD;
    TI:=TI;
    KP:= KP;

```

```

ED:= (c * CONS_NPT11 - PT11V);
ERR:= CONS_NPT11 - PT11V;

UP:= KP * (b * CONS_NPT11 - PT11V);
UI:= UI_1 + (KP * (TEMPS/TI)*ERR);
UD:= ((TD/(TD + TEMPS*N))*UD) + (((KP*TD*N)/(TD+TEMPS*N))*(ED - ED_1));

IF (UI_1 > 10.0) AND (UI_1 < 0.0) THEN
    UI := UI_1;
ELSE
    UI:= UI;
END_IF;

YV:= (UP + UI + UD);
Y:= (UP + UI + UD)* (32767.0/10.0);

PT11V_1:=PT11V;
ED_1:= ED;
UI_1:= UI;

ELSE
    Y:=0;
    ED:=0;
    ED_1:=0;
    PT11V:=0;
    PT11V_1:=0;
    UI:=0;
    UD:=0;
    UP:=0;

END_IF;

//NIVELL DE PRESSIO

IF ((PT11V>= CONS_NPT11-0.05 ) AND (PT11V<= CONS_NPT11+0.05 )) THEN
    N_PT11 := TRUE;
ELSE
    N_PT11 := FALSE;
END_IF;

//NIVELL DEL VAS

IF (((NIVELL_A - CONS_NVAS) >= LT16LOG -16 ) AND ((NIVELL_A - CONS_NVAS) >= LT16LOG
+ 16 )) THEN
    N_VAS := TRUE;
ELSE
    N_VAS := FALSE;
END_IF;

//NIVELL INICIAL

IF ((LT16LOG >= CONS_NINIT -16) AND (LT16LOG <= CONS_NINIT +16)) THEN
    N_INIT := TRUE;
ELSE
    N_INIT := FALSE;
END_IF;

//_____

// TRANSICIONS ENTRADES

//G0

T0:= E13 AND NOT(E112) AND LT4 AND LT3;
T1:= TON_01.Q;
T2A:= N_VAS AND LT3;
T2B:= NOT(LT3);
T3:= N_PT11 ;

```



```

T4:= TON_02.Q;
T5:= (LT16LOG >= NIVELL_A -16) AND (LT16LOG <= NIVELL_A +16) AND LT4 AND LT3;
//T6:= TON_03.Q; //REARME

//G10

T10:= E111 AND E0 AND LT5 AND LT4 AND LT3;
T11:= TON_01.Q;
T12A:= N_INIT AND LT4 AND LT3 AND NOT(LT2);
T12B:= LT2 AND LT3;
T12C:= NOT(LT2) AND NOT(LT3);
T13:= E22;
//T14:= TON_03.Q;
//T15:= TON_03.Q;

//G20

T20:= E112 AND E13 AND E0 AND NOT(LT5) AND LT4 AND LT3 OR E124;
T21:= LT5;
T22:= E10;

//G110

T110:= (VSTART OR START) AND E121;
T111:= (NOT(STOP) OR VSTOP) AND E121;
T112:= E22;

// _____

// EVOLUCIO D ESTATS

//GRAF 0 GENERAL
CASE ESTAT OF
0:

    IF (T0 = TRUE ) THEN
        ESTAT:=1;
    END_IF;

1:

    IF (T1 = TRUE ) THEN
        ESTAT:=2;
    END_IF;

2:

    IF (T2A = TRUE ) THEN
        ESTAT:=3;
    ELSIF (T2B = TRUE) THEN
        ESTAT:=6;
    END_IF;

3:

    IF (T3 = TRUE ) THEN
        ESTAT:=4;
    END_IF;

4:

    IF (T4 = TRUE ) THEN
        ESTAT:=5;
    END_IF;

5:

    IF (T5 = TRUE ) THEN
        ESTAT:=0;
    END_IF;

6:

```

```
        IF (T6 = TRUE ) THEN
            ESTAT:=0;
        END_IF;

    END_CASE;

//GRAF 10 INICI

CASE GRAF_INIT OF
    10:

        IF (T10 = TRUE ) THEN
            GRAF_INIT:=11;
        END_IF;

    11:

        IF (T11 = TRUE ) THEN
            GRAF_INIT:=12;
        END_IF;

    12:

        IF (T12A = TRUE ) THEN
            GRAF_INIT:=13;
        ELSIF (T12B = TRUE) THEN
            GRAF_INIT:= 14;
        ELSIF (T12C = TRUE) THEN
            GRAF_INIT:= 15;
        END_IF;

    13:

        IF (T13 = TRUE ) THEN
            GRAF_INIT:=10;
        END_IF;

    14:

        IF (T14 = TRUE ) THEN
            GRAF_INIT:=10;
        END_IF;

    15:

        IF (T15 = TRUE ) THEN
            GRAF_INIT:=10;
        END_IF;

END_CASE;

//GRAF 20 FINAL

CASE GRAF_FI OF
    20:

        IF (T20 = TRUE ) THEN
            GRAF_FI:=21;
        END_IF;

    21:

        IF (T21 = TRUE ) THEN
            GRAF_FI:=22;
        END_IF;
```

```

22:

    IF (T22 = TRUE ) THEN
        GRAF_FI:=20;
    END_IF;

END_CASE;

//GRAF 110 FINAL

CASE TECLAT OF
    110:

        IF (T110 = TRUE ) THEN
            TECLAT:=111;
        END_IF;

    111:

        IF (T111 = TRUE ) THEN
            TECLAT:=112;
        END_IF;

    112:

        IF (T112 = TRUE ) THEN
            TECLAT:=110;
        END_IF;

END_CASE;

//
//ACTUALITZACIÓ D ESTATS

//GRAF GENERAL (ESTAT)
E0:= FALSE;
E1:= FALSE;
E2:= FALSE;
E3:= FALSE;
E4:= FALSE;
E5:= FALSE;
E6:= FALSE;

IF (ESTAT = 0) THEN
    E0 := TRUE;
ELSIF (ESTAT = 1) THEN
    E1 := TRUE;
ELSIF (ESTAT = 2) THEN
    E2 := TRUE;
ELSIF (ESTAT = 3) THEN
    E3 := TRUE;
ELSIF (ESTAT = 4) THEN
    E4 := TRUE;
ELSIF (ESTAT = 5) THEN
    E5 := TRUE;
ELSIF (ESTAT = 6) THEN
    E6 := TRUE;
ELSE
    END_IF
//GRAF INICI (GRAF_INIT)

E10:= FALSE;
E11:= FALSE;
E12:= FALSE;
E13:= FALSE;

```

```

E14:= FALSE;
E15:= FALSE;

IF (GRAF_INIT = 10) THEN
    E10 := TRUE;
ELSIF (GRAF_INIT = 11) THEN
    E11 := TRUE;
ELSIF (GRAF_INIT = 12) THEN
    E12 := TRUE;
ELSIF (GRAF_INIT = 13) THEN
    E13 := TRUE;
ELSIF (GRAF_INIT = 14) THEN
    E14 := TRUE;
ELSIF (GRAF_INIT = 15) THEN
    E15 := TRUE;
ELSE
    END_IF

//GRAF FI (GRAF_FI)

E20:= FALSE;
E21:= FALSE;
E22:= FALSE;

IF (GRAF_FI = 20) THEN
    E20 := TRUE;
ELSIF (GRAF_FI = 21) THEN
    E21 := TRUE;
ELSIF (GRAF_FI = 22) THEN
    E22 := TRUE;
ELSE
    END_IF

//GRAF TECLAT (TECLAT)

E110:= FALSE;
E111:= FALSE;
E112:= FALSE;

IF (TECLAT = 110) THEN
    E110 := TRUE;
ELSIF (TECLAT = 111) THEN
    E111 := TRUE;
ELSIF (TECLAT = 112) THEN
    E112 := TRUE;
ELSE
    END_IF

//_____

//TEMPORITZADORS
TON_01(IN:=E1 OR E11, PT:=T#1s);           //TEMPS DE TRANSICIÓ
TON_02(IN:=E4, PT:=T#40s);                 //RECIRCULACIÓ



//_____

//SORTIDES
V11:= E1 OR E2 OR E3 OR E4;
V12:= E1 OR E2 OR E5 OR E11 OR E12;
V13:= E5 OR E11 OR E12;
V14:= E11 OR E12;
V_PID:= E3 OR E4;
VP1:= E2 OR E12;
P1:=BOOL_TO_INT(VP1)*32500 + BOOL_TO_INT(V_PID) * REAL_TO_INT(Y);
VV15:= E21;
V15:=BOOL_TO_INT(VV15)*32500;

END_PROGRAM

```

- **Variables locals del programa d'aturada d'Emergència:**

Name	Type
 T121B	BOOL
 T121A	BOOL

- Programa d'aturada d'Emergència PID:

```

PROGRAM _INIT
    EMERGENCIA:= 120;
END_PROGRAM

PROGRAM _CYCLIC

    //CONTROL DE VARIABLES

        //FORÇAT DE LES ETAPES
    IF (E122 = TRUE) THEN
        ESTAT:=0;
        GRAF_INIT:=10;
        GRAF_FI:=20;
        TECLAT:=110;
    ELSE
    END_IF;

    //ALARMA_BLINK
    IF (ALARMA_BLINK = TRUE) THEN
        TCOUNT := TCOUNT + 1;
        IF (TCOUNT =99) THEN
            ALARMA := NOT (ALARMA);
            TCOUNT:=0;
        ELSE
        END_IF;
    ELSIF (NALARMA = TRUE) THEN
        ALARMA := TRUE;
    ELSE
        ALARMA := FALSE;
        TCOUNT:=0;
    END_IF;

    //_____

    // TRANSICIONS ENTRADES
    T120:= OnOff;
    T121A:= E22 OR NOT(OnOff);
    T121B:= NOT(PE) OR E6 OR E14 OR E15;
    T122:= PE;
    T123:= TON_03.Q;
    T124:= E22;

    //_____

    // EVOLUCIO D ESTATS

    CASE EMERGENCIA OF
        120:

            IF (T120 = TRUE ) THEN
                EMERGENCIA:=121;
            END_IF;

        121:

```

```

        IF (T121A = TRUE ) THEN
            EMERGENCIA:=120;
        ELSIF (T121B = TRUE ) THEN
            EMERGENCIA:=122;
        END_IF;

122:

        IF (T122 = TRUE ) THEN
            EMERGENCIA:=123;
        END_IF;

123:

        IF (T123 = TRUE ) THEN
            EMERGENCIA:=124;
        END_IF;

124:

        IF (T124 = TRUE ) THEN
            EMERGENCIA:=120;
        END_IF;

    END_CASE;

//_____

//ACTUALITZACIÓ D ESTATS

E120:= FALSE;
E121:= FALSE;
E122:= FALSE;
E123:= FALSE;
E124:= FALSE;

IF (EMERGENCIA = 120) THEN
    E120 := TRUE;
ELSIF (EMERGENCIA = 121) THEN
    E121 := TRUE;
ELSIF (EMERGENCIA = 122) THEN
    E122 := TRUE;
ELSIF (EMERGENCIA = 123) THEN
    E123 := TRUE;
ELSIF (EMERGENCIA = 124) THEN
    E124 := TRUE;
ELSE
    END_IF;

//_____

//TEMPORITZADORS

TON_03(IN:=((STOP=FALSE) OR (VSTOP=FALSE)), PT:=T#5s);           //REARME ON VSTOP
ES VARIABLE DEL SISTEMA SCADA

//_____

//SORTIDES
NALARMA:= E122;
ALARMA_BLINK:= E123;
POWER:= E121;

END_PROGRAM

```

10.2. Norma de graus de protecció IP

Los **grados de protección IP** hacen referencia a una normativa internacional, que indica el **nivel de protección de los equipos eléctricos o electrónicos frente a la entrada de agentes externos: polvo o agua**.

La **CEI o Comisión Electrotécnica Internacional**, (también conocida como IEC por sus siglas en inglés), establece una normativa para clasificar los diferentes grados de protección de las envolventes de equipos eléctricos y electrónicos frente a agentes externos, especialmente factor humano o agentes medioambientales. O lo que es lo mismo: el grado de protección que tiene el recubrimiento exterior del equipo o dispositivo, frente a la entrada de cuerpos sólidos y líquidos (fundamentalmente polvo o agua).

Para interpretar el grado de protección IP, debes tener en cuenta que:



Así, el **grado más bajo de protección que podemos encontrar es el IP00 y el más alto es el IP68**. Como regla general, podemos decir que cuanto más alto es el grado IP, más protegido y aislado está el equipo frente a la entrada de agentes externos.

Para tener claro qué indica cada valor, lo mejor es que eches un vistazo a estas tablas.

Primer dígito. Hace referencia a la entrada de cuerpos sólidos

NIVEL	DESCRIPCIÓN
0	Sin protección
1	Protegido contra la entrada de elementos sólidos de hasta 50mm.
2	Protegido contra la entrada de elementos sólidos de hasta 12,5mm.
3	Protegido contra la entrada de elementos sólidos de hasta 2,5mm.
4	Protegido contra la entrada de elementos sólidos de hasta 1mm.
5	Protegido contra la entrada de polvo (la cantidad que entra no interfiere con el funcionamiento del dispositivo).
6	Totalmente protegido contra la entrada de polvo.

Segundo dígito. Hace referencia a la entrada de agua

NIVEL	DESCRIPCIÓN
0	Sin protección
1	No debe entrar el agua cuando se la deja caer, desde 200mm de altura respecto del equipo, durante 10 minutos (a razón de 3-5mm³ por minuto).
2	No debe entrar el agua cuando se la deja caer, durante 10 minutos (a razón de 3-5mm³ por minuto). Dicha prueba se realizará 4 veces a razón de una por cada giro de 15° tanto en sentido vertical como horizontal, partiendo cada vez de la posición normal de trabajo.
3	No debe entrar el agua nebulizada en un ángulo de hasta 60° a derecha e izquierda de la vertical a un promedio de 11 litros por minuto y a una presión de 800-100 kN/m² durante un tiempo que no sea menor a 5 minutos.
4	No debe entrar el agua arrojada desde cualquier ángulo a un promedio de 10 litros por minuto y a una presión de 800-100 kN/m² durante un tiempo que no sea menor a 5 minutos.
5	No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión 30 kN/m² durante un tiempo que no sea menor a 3 minutos y a una distancia que no sea menor de 3 metros.
6	No debe entrar el agua arrojada a chorros (desde cualquier ángulo) por medio de una boquilla de 12,5 mm de diámetro, a un promedio de 100 litros por minuto y a una presión 100 kN/m² durante un tiempo que no sea menor a 3 minutos y a una distancia que no sea menor de 3 metros.
7	El equipo debe soportar sin filtración alguna la inmersión completa a 1 metro durante 30 minutos.
8	El equipo debe soportar sin filtración alguna la inmersión completa y continua a la profundidad y durante el tiempo que especifique el fabricante del producto con el acuerdo del cliente, pero siempre que resulten condiciones más severas que las especificadas para el valor 7.

10.3. Implementar i sintonitzar paràmetres d'un PID amb bloc de funcions en Automation Studio

Per configurar el controlador PID es fa servir el bloc de funció “LCRPIDpara ()” o “LCRPIDpara ()”, segons convingui, sempre i quan els paràmetres necessaris per al controlador PID (Kp, Tn, Tv, Tf, ...) ja es coneixen. Aquests dos blocs de funcions tenen d'anar junts ja que un implementa el PID i l'altre el configura.

Taula 22 - Taula de variables utilitzades per definir el bloc de funcions “LCRPID()”.

I/O	Parameter	Data type	Description
IN	enable	BOOL	This function block is only executed if <i>enable</i> = TRUE.
IN	ident	UDINT	Ident of LCRPIDpara() or LCRPIDTune() .
IN	W	REAL	Reference variable (set value).
IN	X	REAL	Controlled variable (actual value).
IN	Y_max	REAL	Upper manipulated variable limit. Permitted values: -100.0..+100.0 [%]. Limits the modulation of <i>Yi</i> and <i>Y</i> to a smaller area than the maximum possible -100% to +100%. Must be greater than <i>Y_min</i> . This input can be changed at any time and immediately affects the controller.
IN	Y_min	REAL	Lower manipulated variable limit. Permitted values: -100.0..+100.0 [%]. Limits the modulation of <i>Yi</i> and <i>Y</i> to a smaller area than the maximum possible -100% to +100%. Must be less than <i>Y_max</i> . This input can be changed at any time and immediately affects the controller.
IN	A	REAL	Feed forward value. Permitted values: -100.0..+100.0 [%]. The feed forward value is added to the proportional, integral and differential element of the manipulated variable.
IN	Y_man	REAL	Manipulated variable for manual operation. Permitted values: -100.0..+100.0 [%]. Manual operation is activated by setting the input <i>mode</i> to LCRPID_MODE_MAN or LCRPID_MODE_MAN_JOLTFREE.
IN	Y_fbk	REAL	Manipulated variable fed back to the windup protection if the parameter <i>fbk_mode</i> in the LCRPIDpara() or LCRPIDTune() function block is set to LCRPID_FBK_MODE_EXTERN (-100.0...+100.0 [%]).
IN	hold_I	BOOL	If this input is TRUE, the I-element is held at its current value.
IN	mode	USINT	Controller operating mode: LCRPID_MODE_OFF: Controller switched off. LCRPID_MODE_AUTO: Automatic operation. LCRPID_MODE_MAN: Manual mode, manipulated variable is set to <i>Y_man</i> . LCRPID_MODE_OPEN: Manipulated variable is set to <i>Y_max</i> .

			<p>LCRPID_MODE_CLOSE: Manipulated variable is set to Y_{min}. LCRPID_MODE_FREEZE: Freezes the manipulated variable.</p> <p>When switching from this operating mode to the operating mode LCRPID_MODE_AUTO, any existing control deviation caused by the P-element will lead to a manipulated variable jump.</p> <p>LCRPID_MODE_MAN_JOLTFREE, LCRPID_MODE_OPEN_JOLTFREE, LCRPID_MODE_CLOSE_JOLTFREE, LCRPID_MODE_FREEZE_JOLTFREE: Switching back to LCRPID_MODE_AUTO takes place jolt-free (i.e. no manipulated variable jump), so long as the I-element is enabled. Any control deviation at the time of switching is successively compensated by the I-element.</p>
OUT	status	UINT	<p>Function block status report</p> <p>LCR_WARN_Tx_DT: Tf was specified as smaller than $1.4 * dt$ (controller scan time, cycle time). Tf is therefore limited to this value.</p> <p>Other: see Error numbers (0 = no error).</p>
OUT	e	REAL	Control deviation. Value range: -100.0..+100.0 [%].
OUT	Y	REAL	Manipulated variable. Value range: -100.0..+100.0 [%].
OUT	Yp	REAL	Proportional manipulated variable element. Value range: -100.0..+100.0 [%].
OUT	Yi	REAL	Integral manipulated variable element. Value range: -100.0..+100.0 [%].
OUT	Yd	REAL	Differential manipulated variable. Value range: -100.0..+100.0 [%].

Taula 23 - Taula de variables utilitzades per definir el bloc de funcions "LCRPIDpara()".

I/O	Parameter	Data type	Description
IN	enable	BOOL	This function block is only executed if <i>enable</i> = TRUE. Otherwise, all outputs (including ident!) are set to 0.
IN	enter	BOOL	Control parameters are only checked and converted into an internal display when this input is TRUE. Otherwise, modifications to the parameter inputs are ignored and computing time can be saved.
IN	WX_max	REAL	<p>Maximum input value of the reference variable W and the controlled variable X. Must be larger than WX_{min}.</p> <p>WX_{min} and WX_{max} are used for scaling the internal calculations processed in the value range between 0% and 100%. This prevents calculation inaccuracies by displaying numbers with the data type REAL.</p>

			<p>If WX_{min} is set to 0.0 and WX_{max} to 100.0, then W and X are not scaled. (Default setting)</p> <p>W and X are not limited to the range defined by WX_{min} and WX_{max}. The calculations will however become less accurate the more they deviate from this value range.</p> <p><u>Warning:</u> As a rule, changing WX_{min} or WX_{max} entails changing the reinforcement on the controller!</p>
IN	WX_min	REAL	<p>Minimum input value of the reference variable W and the controlled variable X. Must be less than WX_{max}.</p> <p>WX_{min} and WX_{max} are used for scaling the internal calculations processed in the value range between 0% and 100%. This prevents calculation inaccuracies by displaying numbers with the data type REAL.</p> <p>If WX_{min} is set to 0.0 and WX_{max} to 100.0, then W and X are not scaled. (Default setting)</p> <p>W and X are not limited to the range defined by WX_{min} and WX_{max}. The calculations will however become less accurate the more they deviate from this value range.</p> <p><u>Warning:</u> As a rule, changing WX_{min} or WX_{max} entails changing the reinforcement on the controller!</p>
IN	invert	BOOL	TRUE inverts the controller's direction of movement and can therefore implement negative gains. A positive control deviation decreases the manipulated variable Y .
IN	deadband	REAL	Dead band. The deadband limits are defined by $(W - deadband)$ and $(W + deadband)$. Permitted values: $0..(WX_{max} - WX_{min})$. Deadband is disabled when $deadband = 0$.
IN	deadband_mode	USINT	<p>Deadband mode:</p> <p>LCRPID_DEADBAND_MODE_ZERO (0): If deadband mode is enabled ($deadband > 0$) and if $(W - deadband) < X < (W + deadband)$, then the controller remains active until $X = W$ or $X = W$. The control deviation is not set to $e = 0$ until the set value W is reached. Controller output Y retains the value that was output when the set value was reached.</p> <p>LCRPID_DEADBAND_MODE_STANDARD (1): If deadband mode is enabled ($deadband > 0$) and if $(W - deadband) < X < (W + deadband)$, the controller deviation e is set to 0 internally. Controller output Y retains the value that was output when the deadband mode began.</p> <p>LCRPID_DEADBAND_MODE_STANDARD_I (2): Same as the mode LCRPID_DEADBAND_MODE_STANDARD, however, after leaving the deadband, the integral element calculates the control deviation e as '$W - X \pm deadband$' instead of '$W - X$'.</p>
IN	dY_max	REAL	Maximum manipulated variable ramp (rate of change of the manipulated variable). Units: % / seconds.

			The ramp is disabled at 0 and does not require any computing time.
IN	Kp	REAL	<p>Proportional gain. Units: [%/%]. Permitted values: $Kp > 0.0$. Negative gains are configured by setting the input <i>invert</i> = TRUE.</p> <p>The control error $e = W - X$ is also scaled when scaling the set value W and the actual value X between WX_{min} and WX_{max}, whereby $e = 100\%$ is equal to a control deviation of $(W - X) = (WX_{max} - WX_{min})$.</p> <p>$Kp$ [%/%] = $k_p * (WX_{max} - WX_{min})/100$ must be defined to implement a proportional gain k_p [%/(unit of W and X)], which is specified as a relationship between the manipulated variable percentage and a control deviation in physical units of the controlled variable.</p> <p>Example for calculating Kp.</p>
IN	Tn	REAL	<p>Integral action time (I-element). Units: Seconds. The time span, which a constant control deviation must meet for the I-element to generate the same manipulated variable as the P-element. The integral element and jolt-free switching are disabled at 0 and do not require any computing time.</p>
IN	Tv	REAL	<p>Derivative action time (D-element). Units: Seconds. The time span, which a increasing control deviation of 0 with a constant gradient must meet for the P-element to generate the same manipulated variable as the D-element. The derivative element is disabled at 0 and does not require any computing time. Otherwise, $Tv > Tf > 1.4 * dt$ (controller scan-time, cycle time) must be the case.</p>
IN	Tf	REAL	<p>Filter time for the derivative element. Units: Seconds. Must be specified if Tv is not equal to 0. If Tf is less than $1.4 * dt$ (controller scan time, cycle time) during runtime, then the LC_WARN_Tx_DT warning is output to LCRPID() and calculation takes place with the replacement value $1.4 * dt$.</p>
IN	Kw	REAL	<p>Set value attenuation. The attenuator is disabled at 1 and does not require any computing time.</p>
IN	Kfbk	REAL	<p>Windup damping. The windup protection is disabled at 0 and does not require any computing time.</p>
IN	fbk_mode	USINT	<p>Feedback mode for the manipulated variable: LCRPID_FBK_MODE_INTERN: Internal. LCRPID_FBK_MODE_EXTERN: External for release control. LCRPID_FBK_MODE_LIMIT_YI: The I element is stopped if $Yp + Yi + Yd \geq Ymax - A$ or $Yp + Yi + Yd \leq Ymin + A$. Kfbk does not affect this type of windup protection.</p>

IN	d_mode	USINT	Differentiator mode: LCRPID_D_MODE_X: Only -X (actual value) LCRPID_D_MODE_E: e (control deviation, as for a standard PID controller)
OUT	status	UINT	Function block status message, see Error Numbers (0 = no error).
OUT	ident	UDINT	Ident for LCRPID() . Address of the parameter structure with the control parameters in internal representation.

Un exemple de programació em text estructurat seria el següent:

```
(* init program *)

(* Init variables *)

setValue      := 100.0;

start         := FALSE;

paraEnter := FALSE;

(* PID parameters *)

LCRPIDpara_0.WX_max := 100.0;

LCRPIDpara_0.WX_min := 0.0;

LCRPIDpara_0.invert := FALSE;

LCRPIDpara_0.deadband := 1.0;

LCRPIDpara_0.dY_max := 0.0;

LCRPIDpara_0.Kp      := 2.7;

LCRPIDpara_0.Tn      := 6.2;

LCRPIDpara_0.Tv      := 1.6;

LCRPIDpara_0.Tf      := 0.16;

LCRPIDpara_0.Kw      := 1.0;

LCRPIDpara_0.Kfbk     := 1.0;          (* windup damping enabled *)

LCRPIDpara_0.fbk_mode := LCRPID_FBK_MODE_INTERN;

LCRPIDpara_0.d_mode   := LCRPID_D_MODE_X;


(* Parameters for PID controller *)

LCRPID_0.Y_max := 100.0;

LCRPID_0.Y_min := -100.0;

LCRPID_0.A      := 0.0;

LCRPID_0.Y_man  := 0.0;

LCRPID_0.Y_fbk  := 0.0;

LCRPID_0.hold_I := FALSE;

LCRPID_0.mode   := LCRPID_MODE_AUTO;
```

```
(* Parameters for PT2 element *)

LCRPT2_0.V      := 3.9;

LCRPT2_0.T1     := 0.9;

LCRPT2_0.T2     := 15.3;

LCRPT2_0.y_set  := 0.0;

LCRPT2_0.set    := 0;

(* cyclic program *)

(* PID parameters *)

LCRPIDpara_0.enable := start;

LCRPIDpara_0.enter  := paraEnter;

LCRPIDpara_0();      (* LCRPIDpara function block call *)

paraEnter := FALSE;

(* PID controller *)

LCRPID_0.enable := start;

LCRPID_0.ident  := LCRPIDpara_0.ident; (* ident of PIDpara -> provides parameters
(Kp, Tn, Tv, ...) *)

LCRPID_0.W      := setValue;

LCRPID_0.X      := actValue;

LCRPID_0();      (* LCRPID function block call *)

manipulatedVar := LCRPID_0.Y;

(* PT2 element *)

LCRPT2_0.enable := start;

LCRPT2_0.x      := manipulatedVar;

LCRPT2_0();      (* LCRPT2 function block call *)

actValue := LCRPT2_0.y;
```

L'element PT2 com a sistema configura un circuit tancat juntament amb el controlador PID, de manera que el valor de sortida de l'element PT2 es torna al controlador com a valor real “actValue”.

“LCRPIDTune ()” està enllaçat amb el corresponent bloc de funció “LCRPID ()” en comptes de fer servir “LCRPIDpara ()” proporcionant tots els paràmetres necessaris. Això evita la configuració manual del controlador amb el bloc de funció “LCRPIDpara ()”. En la taula 24 es mostren totes les variables que tracta el bbloc de funció “LCRPIDpara ()”.

Taula 24 - Taula de variables utilitzades per definir el bloc de funcions “LCRPIDTune()”.

I/O	Parameters	Data type	Description
IN	enable	BOOL	This function block is only executed if <i>enable</i> = TRUE.
IN	Y_min	REAL	Lower limit for the controller's manipulated variable. Must be less than <i>Y_max</i> ! Allowed values: -100.0..+100.0 [%].
IN	Y_max	REAL	Upper limit for the controller's manipulated variable. Must be greater than <i>Y_min</i> ! Allowed values: -100.0..+100.0 [%].
IN	Y0	REAL	Only for step response! Output variable before performing stepping. Must be selected in a way so that the controlled variable is defined close to the operating point. Allowed values: -100.0..+100.0 [%].
IN	Y1	REAL	Only for step response! New manipulated variable after the step. Must be selected in a manner that noticeably changes the controlled variable X. Must be greater than <i>Y0</i> ! Allowed values: -100.0..+100.0 [%].
IN	okToStep	BOOL	Only for step response! Enabled for manipulated variable jump after the settling phase is complete. This input is used to synchronize multiple tuning function blocks for simultaneously tuning multiple control zones. This input must be set with TRUE or the output <i>rdyToStep</i> if synchronization is not necessary.
IN	X0	REAL	Only for oscillation attempts! Controller value for the uncontrolled system

			(e.g. ambient temperature for temperature systems).
IN	X_min	REAL	Allowed minimum value of the controlled variable during tuning. The tuning is terminated if $X < X_{min}$, and the error LCR_ERR_LCRPIDTune_CONTROL_VAR (31560) is output.
IN	X_max	REAL	Allowed maximum value of the controlled variable during tuning. The tuning is terminated if $X > X_{max}$, and the error LCR_ERR_LCRPIDTune_CONTROL_VAR (31560) is output.
IN	P_manualAdjust	UINT	Fine adjustment of the proportional gain. Allowed values: -99..+10000 [%]. The controller behavior can be adjusted manually if necessary. Positive values increase, negative values decrease the controller gain by the corresponding percentage.
IN	I_manualAdjust	UINT	Fine adjustment of the integral element. Allowed values: -100..+10000 [%]. The controller behavior can be adjusted manually if necessary. Positive values accelerate, negative decelerate the integral element by the corresponding percentage. The I ration is shut off at -100%.
IN	D_manualAdjust	UINT	Fine adjustment of the derivative element. Allowed values: -100..+10000 [%]. The controller behavior can be adjusted manually if necessary. Positive values gain, negative lessen the derivative element by the corresponding percentage. The D ratio is shut off at -100%.
IN	request	UDINT	Start of tuning. LCRPID_TUNE_REQU_OFF (0): No tuning active or deactivate, controller is enabled.

			<p>LCRPID_TUNE_REQU_OSCILLATE (1): Tuning with oscillation attempt.</p> <p>LCRPID_TUNE_REQU_STEPRESPONSE (2): Tuning with step response.</p> <p>These requests start the tuning procedure with the respective default options. The specification of differing options is described under Request with specification of options.</p>
IN	pAddPar	lcrpid_tune_ad-dpar_typ (given as a pointer)	<p>Additional parameters that are otherwise specified using the LCRPIDpara() function block.</p> <p>For a description of the parameters, see under data types and constants: lcrpid_tune_add-par_typ.</p> <p>Predefined values are used automatically if an address is not assigned here or if no values are entered in the specified structure.</p>
IN	pOptions_osc	lcr-pid_tune_osc_options_typ (given as a pointer)	<p>Additional parameters for the oscillation attempt</p> <p>For a description of the parameters, see under data types and constants: lcrpid_tune_osc_options_typ.</p> <p>Predefined values are used automatically if an address is not assigned here or if no values are entered in the specified structure.</p>
IN	pOptions_step	lcr-pid_tune_step_options_typ (given as a pointer)	<p>Additional parameters for the step response.</p> <p>For a description of the parameters, see under data types and constants: lcrpid_tune_step_options_typ.</p> <p>The correct setting for this parameter can be a decisive factor for the result of the tuning procedure.</p> <p>Predefined values are used automatically if an address is not assigned here or if no values are entered in the specified structure.</p>
OUT	status	UINT	<p>Function block status message, see Error Numbers (0 = no error).</p>

OUT	addInfo	UINT	For additional information about the status, see Error numbers .
OUT	ident	UDINT	Ident for LCRPID() . Address of the parameter structure with the control parameters in internal representation.
OUT	state	UINT	Step (phase), in which the tuning is currently located. LCRPID_TUNE_STATE_READY (0): Ready to start tuning, no tuning currently active, the controller is enabled. LCRPID_TUNE_STATE_FINISHED (50): Tuning successfully completed. The request input must be set to LCRPID_TUNE_REQU_OFF (0) to reset the function block and enable the controller.
OUT	rdyToStep	BOOL	Only for step response! TRUE if the settling phase is complete and the manipulated variable jump can be performed. This output is used to synchronize multiple tuning function blocks. If synchronization is not necessary, then this output must be returned to the input <i>okToStep</i> .

Un exemple de codi en text estructurat seria el següent:

```
(* init program *)

(* Init variables *)

setValue      := 100.0;

start         := FALSE;

tuningRequest := LCRPID_TUNE_REQU_OFF;

(* Parameters for PID tuning *)

LCRPIDTune_0.Y_min := 0.0;
```

```
LCRPIDTune_0.Y_max      := 100.0;
LCRPIDTune_0.Y0         := 19.2;
LCRPIDTune_0.Y1         := 29.2;
LCRPIDTune_0.X0         := 0.0;
LCRPIDTune_0.X_min      := -10000;
LCRPIDTune_0.X_max      := 10000;
LCRPIDTune_0.P_manualAdjust := 0;
LCRPIDTune_0.I_manualAdjust := 0;
LCRPIDTune_0.D_manualAdjust := 0;
LCRPIDTune_0.pAddPar     := ADR(addParameter);
LCRPIDTune_0.pOptions_osc := ADR(oscOptions);
LCRPIDTune_0.pOptions_step := ADR(stepOptions);

(* Parameters for PID controller *)
LCRPID_0.Y_max := LCRPIDTune_0.Y_max;
LCRPID_0.Y_min := LCRPIDTune_0.Y_min;
LCRPID_0.A      := 0.0;
LCRPID_0.Y_man  := 0.0;
LCRPID_0.Y_fbk  := 0.0;
LCRPID_0.hold_I := FALSE;
LCRPID_0.mode   := LCRPID_MODE_AUTO;

(* Parameters for extruder simulation model *)
extruderLCRSimModExt.enable := TRUE;
extruderLCRSimModExt.Tt_h   := 900000; (* microseconds *)
extruderLCRSimModExt.Tt_c   := 500000; (* microseconds *)
extruderLCRSimModExt.k_h    := 3.9;
extruderLCRSimModExt.k_c    := 0.012;
extruderLCRSimModExt.PT2_T1 := 5.0;
extruderLCRSimModExt.PT2_T2 := 12.5;
```

```
extruderLCRSimModExt.Temp_amb := 25.0;

extruderLCRSimModExt.Temp_c   := 25.0;

extruderLCRSimModExt.Alpha_c  := 0.0;

(* cyclic program *)

(* PID tuning *)

LCRPIDTune_0.enable           := start;

LCRPIDTune_0.okToStep := LCRPIDTune_0.rdyToStep;

LCRPIDTune_0.request          := tuningRequest;

LCRPIDTune_0();               (* LCRPIDTune function block call *)

IF LCRPIDTune_0.state = LCRPID_TUNE_STATE_FINISHED THEN

    tuningRequest := LCRPID_TUNE_REQU_OFF;

END_IF

(* PID controller *)

LCRPID_0.enable := start;

LCRPID_0.ident  := LCRPIDTune_0.ident; (* ident of PIDTune -> provides parameters
(Kp, Tn, Tv, ...) *)

LCRPID_0.W      := setValue;

LCRPID_0.X      := actValue;

LCRPID_0();      (* LCRPID function block call *)

manipulatedVar := LCRPID_0.Y;

(* Extruder simulation model *)

extruderLCRSimModExt.Alpha_h := manipulatedVar;

extruderLCRSimModExt();      (* LCRSimModExt function block call *)

actValue := extruderLCRSimModExt.y;
```